C3oT: Generating Shorter Chain-of-Thought Without Compromising Effectiveness

Yu Kang, Xianghui Sun, Liangyu Chen *, Wei Zou

Beike Inc., Beijing, China
{kangyu009, sunxianghui002, chenliangyu003, zouwei026}@ke.com

Abstract

Generating Chain-of-Thought (CoT) before deriving the answer can effectively improve the reasoning capabilities of large language models (LLMs) and significantly improve the accuracy of the generated answer. However, in most cases, the length of the generated CoT is much longer than the desired final answer, which results in additional decoding costs. Furthermore, existing research has discovered that shortening the reasoning steps in CoT, even while preserving the key information, diminishes LLMs' abilities. These phenomena make it difficult to use LLMs and CoT in many real-world applications that only require the final answer and are sensitive to latency, such as search and recommendation. To reduce the costs of model decoding and shorten the length of the generated CoT, this paper presents **C**onditioned **C**ompressed **C**hain-of-**T**hought (C3oT), a CoT compression framework that involves a compressor to compress an original longer CoT into a shorter CoT while maintaining key information and interpretability, a conditioned training method to train LLMs with both longer CoT and shorter CoT simultaneously to learn the corresponding relationships between them, and a conditioned inference method to gain the reasoning ability learned from longer CoT by generating shorter CoT. We conduct experiments over four datasets from arithmetic and commonsense scenarios, showing that the proposed method is capable of compressing the length of generated CoT by up to more than 50% without compromising its effectiveness.

Introduction

The Chain-of-Thought (CoT) (Nye et al. 2021; Marasović et al. 2021; Wei et al. 2022; Kojima et al. 2022; Lampinen et al. 2022) methodology significantly augments the reasoning abilities of large language models (LLMs), providing critical capabilities for sub-task decomposition in complex problem-solving scenarios. Furthermore, models trained with rich signals, including reasoning processes; explanation traces; and step-by-step thought processes, generally exhibit superior performance (Mukherjee et al. 2023; Mitra et al. 2023). While answering after thinking can elicit highly effective generations by activating LLMs' reasoning abilities, the intermediate reasoning steps in model outputs are often much longer than the desired final answers, notably increasing the cost during the inference phase, and hindering the model's employment in many real-world applications, such as search and recommendation, which usually focus only on the final answer and is sensitive to latency. Therefore, striking a balance between the demand for fast decoding in LLMs applications and the need for long reasoning steps has become an urgent issue.

However, recent studies indicate that lengthening the reasoning steps in CoT considerably enhances LLMs' reasoning abilities across multiple tasks. Alternatively, shortening the reasoning steps, even while preserving the key information, significantly diminishes the reasoning abilities of models (Jin et al. 2024). Fu et al. (2022) propose a complexity-based method for CoT selection and find that CoT with higher reasoning complexity, i.e., chains with more reasoning steps, achieve substantially better performance on multi-step reasoning tasks. Similar conclusions have been drawn from Merrill and Sabharwal (2023)'s work, which explores the relationship between the capabilities of LLMs and the number of CoTs' reasoning steps, varying from logarithmic, linear, to polynomial, based on input length. They also found that the increase in LLMs' computational power depends crucially on the amount of intermediate reasoning steps added.

There has been little work (Deng et al. 2023; Liu et al. 2024) focused on compressing the length of generated CoT without sacrificing model performance. Implicit-CoT (Deng et al. 2023) attempted to use LLMs' internal hidden states to perform implicit reasoning, replacing explicitly producing the CoT reasoning steps, but the results of this method is still significantly falling behind the explicit CoT method.

Based on these results, we ask: *Is there a method that can significantly reduce the length of intermediate reasoning steps in generated CoT without compromising effectiveness?* The answer is *yes*, we propose **C**onditioned **C**ompressed **C**hain-of-**T**hought (C3oT), a CoT compression framework, to achieve this goal. Specifically, we first present a CoT compressor to condense the original complex CoT into their shortest form while retaining essential information and interpretability, now we have pairs of longer CoT and shorter CoT. We further introduce a conditioned training method to train LLMs with both longer CoT and shorter CoT simultaneously, and by conditioning longer CoT and shorter CoT using distinct initial prompt tokens before instructions, LLMs can

*The corresponding author.

learn the differences and connections between them. Lastly, we propose the conditioned inference method which is used in the inference phase, by applying the initial prompt tokens used for conditioning the shorter CoT before instructions, LLMs can generate CoT with significantly shorter length during inference while maintaining the accuracy of the derived final answer.

To validate the effectiveness of our approach, we conduct experiments on four datasets from two domains that require reasoning, i.e., arithmetic (GSM8K, MathQA) and commonsense (ECQA, StrategyQA). The results show that our method's performance is on par with models trained using only the original longer CoT across all datasets, while significantly shortening the length of generated CoT. Additionally, we design extensive experiments and discussions to analyze the contribution of different components in our approach, as well as to explore future research directions of CoT compression based on our method.

The contributions of this paper are as follows:

- We propose C3oT, a CoT compression framework used to reduce the cost of model inference by drastically shortening the length of CoT generated by LLMs without loss of effectiveness. We are the first to significantly shorten the length of CoT in model outputs without sacrificing performance, filling the gap in the field of model inference acceleration in terms of shortening the length of intermediate output.
- Comprehensive experiments demonstrate that our CoT compression method outperforms all baselines on various reasoning tasks, such as math reasoning (GSM8K, MathQA) and commonsense reasoning (ECQA, StrategyQA). We conduct detailed ablation studies and analyses to prove the effectiveness of our approach.
- We conduct a series of extension experiments based on the proposed C3oT framework, providing insights for future research directions in the field of CoT compression and further demonstrating the effectiveness of our method.

Related Work

LLMs Inference Acceleration

Due to the conflict between the outstanding performance of LLMs and the difficulty of their application in real-world scenarios, an increasing amount of research is focusing on accelerating the inference of LLMs. These works primarily focus on reducing the number of input tokens to be processed in order to reduce the inference cost. Some approaches focus on reducing the length of prompts by using prompt tuning to learn special tokens (Wingate, Shoeybi, and Sorensen 2022; Chevalier et al. 2023; Ge et al. 2023; Mu, Li, and Goodman 2024). Some approaches attempt to compress prompt based on information entropy (Li et al. 2023; Jiang et al. 2023a,b) and data distillation (Pan et al. 2024). Some studies utilize LLMs to summarize input dialog or data, transforming them into efficient memory and knowledge (Chase 2022; Zhang et al. 2023). There are also some studies focus on token pruning or token merging (Kim et al. 2022; Modarressi, Mohebbi, and Pilehvar 2022; Bolya et al. 2022).

However, during the inference stage, the cost of decoding and generating output by the model is significantly higher than the cost of processing the input. Therefore, as the enhancements of model capabilities brought by CoT gain increasing attention, the additional cost involved in generating CoT cannot be ignored. Nevertheless, accelerating the generation of CoT has not received widespread attention.

Recently, only Implicit-CoT (Deng et al. 2023) has attempted to accelerate the generation of CoT. It uses the hidden states of different layers in LLMs to perform implicit reasoning based on knowledge distillation, avoiding the explicit generation of CoT and thereby accelerating the inference process. But Implicit-CoT severely sacrifices model performance. The results generated by this method significantly fall behind those of the explicit CoT method.

The method proposed in this paper also aims to accelerate the inference by reducing the length of generated CoT. By utilizing conditioned training method, the model is enabled to learn both longer and shorter CoT simultaneously, and during the conditioned inference phase, the model is able to stimulate the reasoning capabilities learned from the longer CoT by generating a shorter CoT. In this way, our approach significantly reduces the length of generated CoT without compromising the effectiveness of the model.

It's worth mentioning that there is a line of studies that attempt to accelerate inference through model quantization (Dettmers et al. 2022; Frantar et al. 2022; Xiao et al. 2023), pruning (Frantar and Alistarh 2023; Sun et al. 2023; Das, Ma, and Shen 2023), and other similar techniques. These methods are orthogonal to ours and can be used together.

Chain-of-Thought Analyzing

There is some research focusing on exploring the relationship between the length of CoT and its effects. Interestingly, all of these studies (Fu et al. 2022; Merrill and Sabharwal 2023; Jin et al. 2024) have found that lengthening the intermediate reasoning steps in the CoT can enhance LLMs' capabilities. Fu et al. (2022) find that CoT with higher reasoning complexity, i.e., chains with more reasoning steps, achieve substantially better performance on multi-step reasoning tasks. Merrill and Sabharwal (2023) explore the relationship between the capabilities of LLMs and the number of CoTs' reasoning steps, varying from logarithmic, linear, to polynomial, based on input length, and they find that the increase in LLMs' computational power depends crucially on the amount of intermediate reasoning steps added. Jin et al. (2024) design experiments that expand and compress the reasoning steps in CoT while keeping all other factors constant. They find that lengthening the reasoning steps even without adding new information considerably enhances LLMs' reasoning abilities. Conversely, shortening the reasoning steps while preserving the key information, significantly diminishes the reasoning abilities of models.

This paper also focuses on the relationship between the length of CoT and its effectiveness, but we propose a method that can significantly compress the length of generated CoT without compromising its effectiveness.

Method

Problem Statement

Given a dataset $\{(x_i, r_i^{long}, y_i)\}_{i=1}^N$, where x_i denotes the instruction, y_i is its corresponding answer and r_i^{long} is a well-designed detailed CoT for deriving the answer. We consider a compressor $\mathcal{F}$ that systematically compress any input CoT to its shortest form $r^{short} = \mathcal{F}(r^{long})$, retaining only the key information. Our goal is to train an LLM on $\mathcal{D} = \{(x_i, r_i^{long}, r_i^{short}, y_i)\}_{i=1}^N$ so that during inference, the distribution of generated answers derived from compressed, shorter CoT is as similar to answers derived from original, longer CoT as possible.

Conditioned Compressed Chain-of-Thought (C3oT)

Next, we elaborate on the details of the C3oT framework, which shortens the generated CoT during inference without compromising its effectiveness. An overview of the proposed framework is shown in Figure 1.

Compressor The CoT compressor can be any summarization model that processes the input text to only retain its core information and returns a condensed version. In this paper, we employ GPT-4 (Achiam et al. 2023) as the compressor for CoT compression.

Using GPT-4 as the compressor is also to validate the conclusions of previous research. Specifically, even when using the current most powerful closed-source model to ensure that the compressed CoT retains all key information and interpretability while merely removing redundant words, only using these compressed, shorter CoT to derive the answers will still affect the model's performance. This is consistent with the conclusions of previous research and further proves the value of our approach.

We prompt GPT-4 to obtain the corresponding compressed, shorter CoT r_i^{short} for all original, longer CoT r_i^{long} in the dataset $\{(x_i, r_i^{long}, y_i)\}_{i=1}^N$, and compose $\mathcal{D} = \{(x_i, r_i^{long}, r_i^{short}, y_i)\}_{i=1}^N$. We also investigate the impact on our approach of employing different models as compressors in the Analysis section.

Conditioned Training Inspired by OpenChat (Wang et al. 2023), we can regard $\mathcal{D}$ as a class-conditioned dataset $\mathcal{D}_c = \{(x_i, \tilde{r}_i, y_i, c_i)\}$. Each instruction x_i in the dataset corresponds to both a longer CoT and a shorter CoT. The CoT of different lengths are distinguished through different conditions:

$$\tilde{r}_i = \begin{cases} r_i^{long} & \text{if } c_i = long \\ r_i^{short} & \text{if } c_i = short \end{cases}$$

where $\tilde{r}_i$ can be either r_i^{long} or r_i^{short}, controlled by condition c_i.

Then we fine-tune an LLM on $\mathcal{D}_c$ to teach it the relationship between r_i^{long} and r_i^{short}. We model the LLM to be fine-tuned as a class-conditioned policy $\pi_\theta(y, \tilde{r}|x, c)$. This can be easily implemented by conditioning each CoT which belongs to either a longer CoT or a shorter CoT using distinct initial prompt tokens before instruction as shown below:

[Long Condition]	Answer and provide a detailed thought process:
[Short Condition]	Answer and provide as brief a thought process as possible:

After adding conditions in this way, each sample in the original dataset, for example:

Instruction: *Natalia sold clips ... How many clips did Natalia sell altogether in April and May?*
Rationale: *... Natalia sold 48+24 = «48+24=72»72 clips altogether in April and May.*

becomes two samples containing the original, longer CoT and the compressed, shorter CoT, respectively:

Instruction: *Answer and provide a detailed thought process: Natalia sold clips ... How many clips did Natalia sell altogether in April and May?*
Rationale: *... Natalia sold 48+24 = «48+24=72»72 clips altogether in April and May.*

and

Instruction: *Answer and provide as brief a thought process as possible: Natalia sold clips ... How many clips did Natalia sell altogether in April and May?*
Rationale: *... She sold 72 clips in April and May.*

Now, fine-tuning an LLM on $\mathcal{D}_c$ is the same as general supervised fine-tuning (SFT). It is worth mentioning that during fine-tuning, longer CoT and shorter CoT in $\mathcal{D}_c$ do not need to appear in pairs, and there is no need to use any method to inform the LLM how they correspond to each other, just randomly shuffle $\mathcal{D}_c$ and train the model like general SFT. Additionally, our conditioned training method can also be regarded as a data augmentation method, but it does not introduce any extra knowledge (Maini et al. 2024).

Conditioned Inference During the inference phase, we assume that the model after conditioned training has learned the differences and connections between the longer CoT and the shorter CoT. Considering that we aim to apply the fine-tuned model to a real-world application and exclusively generate shorter CoT to derive the needed final answer efficiently, we use the same specific prompts that were employed in shorter CoT during the conditioned training phase as below:

[Inference Prompt]	Answer and provide as brief a thought process as possible: <Question>

Experiment

Settings

Datasets To comprehensively validate the effectiveness of C3oT, we evaluated its performance across four datasets from two domains. For math reasoning, we use **GSM8K** (Cobbe et al. 2021) and **MathQA** (Amini et al. 2019). As for commonsense reasoning, we use **ECQA** (Aggarwal et al. 2021) and **StrategyQA** (Geva et al. 2021). All these datasets not only contain the final answers but also include the carefully human-designed CoT used to arrive at the final answers. We

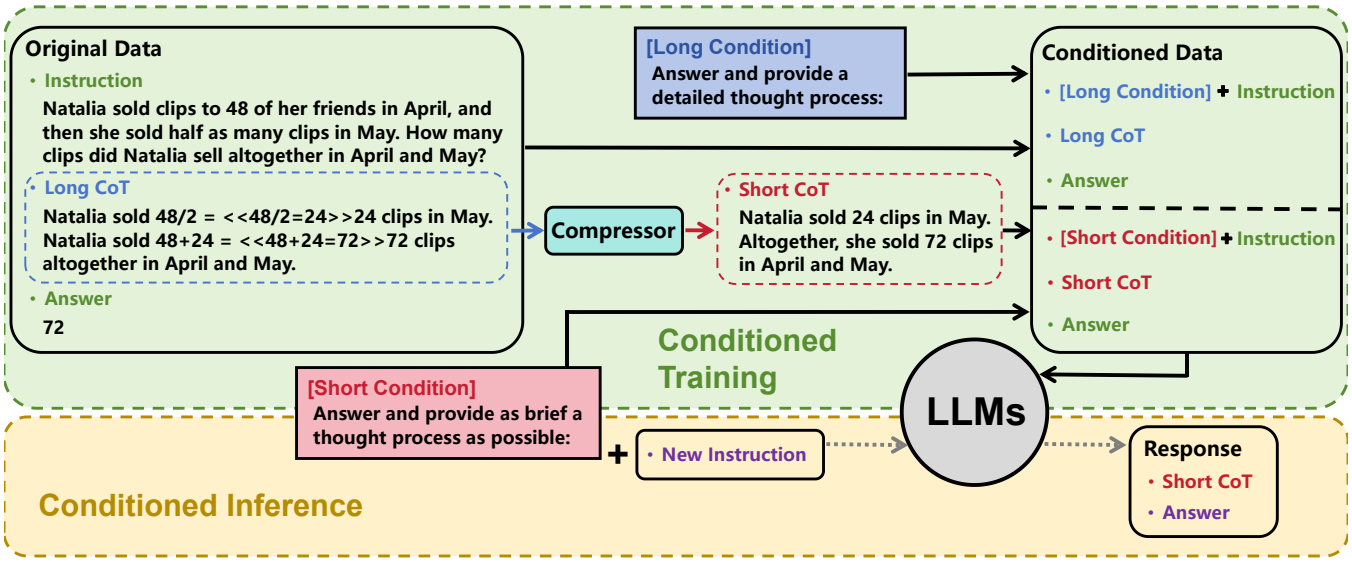


Figure 1: The overall framework of C3oT. Solid arrows denote the conditioned training phase, while dashed arrows denote the conditioned inference phase.

followed the training and testing set division as outlined in the original paper of the dataset used, trained C3oT on the training set, and evaluated its performance on the test set, excluding StrategyQA. Due to the inaccessibility of ground truths for the StrategyQA test set, we proceeded to further split the original StrategyQA training set into training and test sets.

Implementation Details In this paper, we train C3oT based on `LLaMA-2-Chat-7B` and `-13B` (Touvron et al. 2023). We fine-tune the model for 2 epochs on each dataset using the AdamW optimizer with a sequence length of 2,048 tokens and a batch size of 128. The AdamW optimizer's hyperparameters are set as follows: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-6}$, and weight decay of 0.001. We employ a cosine learning rate schedule with a maximum learning rate of 1×10^{-5}.

Baselines We consider the following baselines:

- *Short CoT*: Use GPT-4 as the compressor to compress the original, longer CoT as much as possible while retaining key information and interpretability, then train models using only these compressed, shorter CoT.
- *Long CoT*: Use only the original, longer CoT to train models.
- *Implicit-CoT* (Deng et al. 2023)[1]: Use the LLM's internal hidden states to perform implicit reasoning, instead of explicitly producing the CoT reasoning steps. The implicit reasoning steps are distilled from a teacher model trained on explicit CoT reasoning.

All the baselines are also trained based on `LLaMA-2-Chat-7B` and `-13B`, including the teacher model and student model of Implicit-CoT. The hyperparameters used for training baselines are the same as those in training C3oT.

[1]For a fair comparison, we do not use the data synthesis methods in the original paper.

Evaluation We measure the following metrics:

- *Accuracy*: Following previous works (Suzgun et al. 2022; Cobbe et al. 2021; Amini et al. 2019; Aggarwal et al. 2021; Geva et al. 2021), we measure accuracy using exact match, computed by comparing the generated final answer with the ground-truth label.
- *Compression Rate*: Additionally, we measure the compression rate to evaluate the reduction in length of the generated CoT. The compression rate ρ is defined as $\rho = (L - \tilde{L})/L$, $\rho \in (-\infty, 1]$, where L is the length of the CoT generated by *Long CoT*, which we regard as the benchmark length. And $\tilde{L}$ is the length of the generated compressed CoT. A larger value of compression rate implies a greater reduction in length, resulting in a lower inference cost, which is preferable. When no intermediate CoT steps are generated, the compression rate can reach 1. If the generated CoT is even longer than L, the compression rate becomes negative.

Main Results

Table 1 reports the results of our approach alongside baselines on GSM8K, MathQA, ECQA and StrategyQA. It can be seen that our proposed C3oT consistently outperforms the Implicit-CoT in accuracy by a large margin in all experiments. Implicit-CoT successfully avoids explicitly generating CoT, achieving a 100% compression rate. However, it significantly sacrifices the model's performance, which is not what we want. In addition, three other conclusions can be drawn from these results:

Firstly, comparing the results of *Short CoT* and *Long CoT* reveals that even while using the most powerful model GPT-4 as the compressor to preserve the key information and interpretability in the compressed, shorter CoT in the training

Model Size	Method	Arithmetic				Commonsense			
		GSM8K		MathQA		ECQA		StrategyQA	
		Acc	Compression Rate	Acc	Compression Rate	Acc	Compression Rate	Acc	Compression Rate
7B	Short CoT	31.01	58.63	46.16	29.53	61.93	53.41	67.59	37.99
	Long CoT	**37.38**	0	**51.46**	0	63.96	0	69.66	0
	Implicit-CoT	11.16	100	14.62	100	21.14	100	30.01	100
	C3oT	**36.92**	56.67	50.35	27.39	**69.38**	51.55	**72.41**	42.04
13B	Short CoT	42.46	59.52	52.97	29.85	66.79	55.27	74.83	47.79
	Long CoT	**48.07**	0	**56.21**	0	68.92	0	**76.21**	0
	Implicit-CoT	14.36	100	17.00	100	23.54	100	35.77	100
	C3oT	**47.10**	57.78	**56.62**	31.04	**71.93**	55.28	**76.55**	44.56

Table 1: The Accuracy (%) and Compression Rate (%) performance of the proposed C3oT and baselines. The **bold** scores denote the best performance, as well as performances within 1% of the best.

sets, it still significantly diminishes the model's effectiveness. This conclusion is consistent with previous studies.

Secondly, while shortening the length of generated CoT reduces the model's performance across all datasets, the degree of performance decrease varies across different datasets. Tasks requiring more reasoning abilities, such as math, experience a greater decrease in performance, whereas tasks with lower reasoning demands, such as commonsense, see a relatively smaller decrease. Similarly, although *C3oT* has achieved similar or even better performance than *Long CoT* on all datasets, there is still a slight lag in mathematical tasks, while in commonsense tasks, it even surpasses or significantly outperforms *Long CoT*. This is still related to the reasoning ability required for the tasks.

Lastly, the compression rates on four datasets show that when preserving the key information and interpretability in the compressed, shorter CoT in the training sets, and keeping the compressor unchanged, the compression rate is only related to the dataset itself and not significantly influenced by the domain of the task. In other words, if the original, longer CoT in the training set is more detailed, containing more redundant information, then the compression rate achievable by *C3oT* will be higher. Additionally, we further analyze the impact of different compressors on compression rates in the Analysis section.

Analysis

In this section, we conduct experiments to answer the following questions, in order to analyze the contributions of different components in our approach, and further explore more future research directions for CoT compression based on the proposed C3oT framework.

What is the contribution of the class-conditioned policy?

We conduct an ablation study on the conditions of C3oT to ascertain the contribution of the class-conditioned policy. For *w/o condition*, we remove the distinct initial prompt tokens before instruction and treat the data containing longer CoT and shorter CoT as equivalent, and then fine-tune the models in the regular supervised fine-tuning (SFT) manner.

Table 2 shows that *C3oT* outperforms *w/o condition* in terms of both accuracy and compression rate across all datasets. This is because without the class-conditioned policy, language models lack explicit signals to discern between the longer CoT and the shorter CoT, and during training, the models not only fail to learn the differences and connections between the longer CoT and the shorter CoT, but also get confused by two kinds of CoT with significantly different lengths, thus affecting the performance.

What is the impact of different compressors on C3oT?

To investigate the impact of different compressors on our method, we conduct experiments comparing the results of using GPT-4 as the compressor with the results of using the open-source models `LLaMA-2-Chat-7B` and `-13B` as the compressors. To distinguish the results of different compressors, in Table 3, we name the results as *Short CoT*$_{<Compressor>}$ and *C3oT*$_{<Compressor>}$. It is worth mentioning that *Short CoT*$_{GPT-4}$ and *C3oT*$_{GPT-4}$ are precisely *Short CoT* and *C3oT* in Table 1. The prompts used for the different compressors are the same.

Comparing the results of *Short CoT*$_{GPT-4}$ and *Short CoT*$_{LLaMA2\_*}$ as well as *C3oT*$_{GPT-4}$ and *C3oT*$_{LLaMA2\_*}$ in Table 3, it's evident that the compressed, shorter CoT generated by LLaMA-2 in the training set are slightly inferior in quality to those generated by GPT-4, resulting in a minor decrease in model accuracy, though not significantly. However, the conciseness of the compressed, shorter CoT generated by LLaMA-2 in the training set are noticeably poorer than those generated by GPT-4, leading to a compression rate that is over 15% lower. This indicates that preserving the key information in the original, longer CoT is not difficult for the compressor, but the more powerful the compressor, the more it can produce concise compressed CoT.

Is C3oT effective for expanded CoT?

Previous studies have shown that lengthening the reasoning steps in CoT can enhance the model's reasoning abilities. Therefore, in this part, we explore whether our approach can

Model Size	Method	Arithmetic				Commonsense			
		GSM8K		MathQA		ECQA		StrategyQA	
		Acc	Compression Rate	Acc	Compression Rate	Acc	Compression Rate	Acc	Compression Rate
7B	C3oT	**36.92**	**56.67**	**50.35**	**27.39**	**69.38**	**51.55**	**72.41**	**42.04**
	w/o condition	34.50	35.59	48.07	15.86	66.93	20.88	70.00	9.51
13B	C3oT	**47.10**	**57.78**	**56.62**	**31.04**	**71.93**	**55.28**	**76.55**	**44.56**
	w/o condition	44.50	40.51	55.34	18.53	70.54	27.19	73.10	20.37

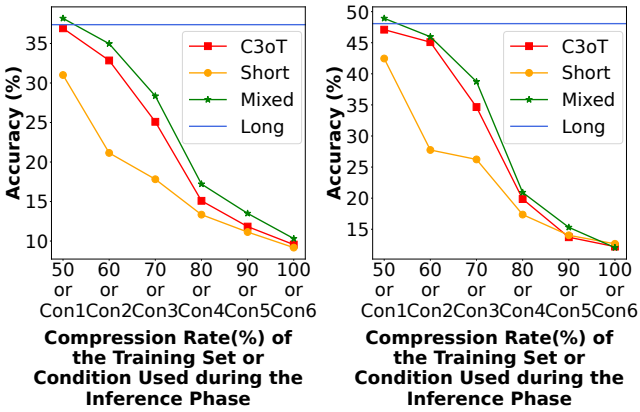
Table 2: Ablation study of the class-conditioned policy (condition) to C3oT.

Model Size	Method	GSM8K	
		Acc	Compression Rate
7B	Long CoT	37.38	0
	Short CoT$_{GPT-4}$	31.01	58.63
	C3oT$_{GPT-4}$	36.92	56.67
	Short CoT$_{LLaMA2_7B}$	31.54	42.28
	C3oT$_{LLaMA2_7B}$	36.13	40.82
	Expanded CoT	39.12	-310.17
	C3oT$_{Expansion}$	37.30	59.27
	C3oT$_{Adapt}$	40.85	70.80
13B	Long CoT	48.07	0
	Short CoT$_{GPT-4}$	42.46	59.52
	C3oT$_{GPT-4}$	47.1	57.78
	Short CoT$_{LLaMA2_13B}$	40.71	40.34
	C3oT$_{LLaMA2_13B}$	46.53	42.48
	Expanded CoT	49.66	-307.88
	C3oT$_{Expansion}$	48.12	57.66
	C3oT$_{Adapt}$	51.09	77.97

Table 3: Performance of some experiments based on C3oT on GSM8K. The negative compression rates represent the degree of increase in length.

compress the much longer, expanded CoT and maintain its effectiveness. We follow the 5 reasoning steps expansion methods proposed by Jin et al. (2024) and use GPT-4 to expand the original CoT. The much longer, expanded CoT is then combined with the compressed, shorter CoT generated by GPT-4 that we used above to form a class-conditioned dataset for training C3oT, which we name *C3oT*$_{Expansion}$. In parallel, we refer to the results obtained from the model trained using only the much longer, expanded CoT as *Expanded CoT*.

Comparing the results of *Long CoT* and *Expanded CoT* in Table 3, it is evident that lengthening the reasoning steps in CoT does improve the model's reasoning abilities, which is consistent with previous studies. While comparing the results of *C3oT*$_{GPT-4}$ and *C3oT*$_{Expansion}$, we observe that although the improvement is not as significant as from *Long CoT* to *Expanded CoT*, *C3oT*$_{Expansion}$ still manages to



(a) Model Size: 7B (b) Model Size: 13B

Figure 2: Accuracy of different methods vs compression rate of the training set or condition used during the inference phase on GSM8K.

achieve a better result than *C3oT*$_{GPT-4}$ while maintaining a similar compression rate. This not only demonstrates the effectiveness of our approach on the much longer, expanded CoT but also presents a method to enhance the model's reasoning abilities without incurring additional costs.

What is the impact of training sets with different compression rates on C3oT?

When we use GPT-4 as the compressor in the previous sections, we prompt it to retain as much key information and interpretability from the original CoT in the training set as possible, the compression rate of the compressed CoT in the resulting GSM8K training set is about 50% compared to the original CoT. We wonder about the impact on model performance when all restrictions on the compressor are removed, and it is only required to compress the original CoT in the training set to a specified compression rate. Specifically, we compress the original CoT in the GSM8K training set with compression rates varying from 50% to 100% (no CoT) in 10% increments.

The results are shown in Figure 2a and Figure 2b. Firstly, we observe that the accuracy of *C3oT* decreases as the compression rate of the training set increases, indicating that *C3oT* is not effective across all compression levels. Secondly,

although *C3oT* outperforms *Short CoT* at almost all training set compression rates, the performance gap between the two widens as the compression rate decreases, only to narrow again at the 50%.

These two phenomena indicate that *C3oT* can only achieve results comparable to *Long CoT* if the compressed, shorter CoT in the training set retains sufficient key information, and if the CoT in the training set is over-compressed, using *C3oT* will still lead to a decline in performance. Furthermore, at high compression rates, the compressed, shorter CoT even loses its grammatical structure, rendering it completely uninterpretable. At this point, the results of *C3oT* and *Short CoT* are not significantly different. As the compression rate of the training set decreases, the information and interpretability contained in the compressed CoT increase. Gradually, *C3oT* can leverage the shorter CoT to activate the reasoning abilities learned during the conditioned training phase from longer CoT, thereby widening the performance gap with *Short CoT*. This continues until the information and interpretability in the compressed, shorter CoT reach a certain threshold, satisfying the requirements for *C3oT* to fully activate the CoT's capabilities, achieving results close to those of *Long CoT*. At the same time, the performance of *Short CoT* also sees a significant improvement.

What is the impact of mixed conditions training on C3oT?

In the previous part, we explored the impact of different compression rates of the training set on C3oT. Specifically, during the training phase, for shorter CoT corresponding to various compression rates, we combined them respectively with the longer CoT to form the class-conditioned dataset. Then, we employed the conditioned training method mentioned in the Method section. However, an intuitive idea is to use various distinct initial prompt tokens before instructions to condition shorter CoT corresponding to different compression rates in the training set, and combine them together with the longer CoT to form a mixed class-conditioned dataset. To investigate this, we implement *Mixed Conditions*. Specifically, we expand the conditions into the following form:

[Long Condition]	Answer and provide a detailed thought process:
[Short Cond.1]	Answer and provide a thought process in compression level of 1:
...	
[Short Cond.6]	Answer and provide a thought process in compression level of 6:

where Short Cond.1 to Cond.6 denote compression rates of the training set ranging from 50% to 100%, respectively.

From Figure 2a and Figure 2b, we can see that *Mixed Conditions* outperforms *C3oT* across all training set compression rates and even surpasses *Long CoT* at 50% compression rate. This demonstrates that training with a mix of data at various compression levels through the class-conditioned policy can lead to mutually beneficial effects.

Moreover, *Mixed Conditions* can generate CoT with different compression levels in the conditioned inference phase by using different initial prompt tokens before instruction. It is worth mentioning that in the Figure 2a and Figure 2b, the horizontal axis represents the compression rate of the training set for *C3oT*, and for *Mixed Conditions*, it represents the distinct initial prompt tokens before instruction corresponding to that compression rate used during the conditioned inference.

Can C3oT select the appropriate compression rate on its own?

Through the exploration of the previous two parts, we've observed that the length of CoT required to correctly answer questions varies with the difficulty of the questions. For simple questions, the model can provide answers directly even without the CoT (Compression Rate = 100%). However, as the questions become more challenging, the model needs to undergo a more complex intermediate reasoning process to arrive at the correct answer. Therefore, the overall accuracy of the model tends to decrease as the compression rate increases. In this final part, we aim to explore the capability of C3oT to automatically select the most appropriate compression rate. For the most appropriate compression rate, we refer to the highest CoT compression rate at which the model still can accurately answer questions.

Inspired by Orca 2 (Mitra et al. 2023), firstly, we arrange the training sets obtained in previous parts by their compression rates, from highest to lowest. Then, we process each training set in order, randomly dividing each into five parts. We use compressed, shorter CoT in four parts to train a model and predict outcomes on the remaining part. This step is repeated three times with different random seeds to obtain three prediction results for each sample in the training set. For each sample, if it is correctly predicted in at least one out of these three attempts, we consider the sample as correctly answerable by the model under the current compression rate and include it in the final training set. Conversely, if a sample is too difficult to be correctly predicted under the current compression rate, it is carried forward to the next round at a lower compression rate for further assessment. Finally, we train C3oT using the conditioned training method mentioned in the Method section with the final training set composed in the above steps and name the result $C3oT_{Adapt}$ in Table 3.

The results from Table 3 show that $C3oT_{Adapt}$ significantly outperforms $C3oT_{GPT-4}$ in both accuracy and compression rate. This demonstrates that after training C3oT with data at the most appropriate compression rate, the model has learned to autonomously determine the most efficient length of the CoT for questions of varying complexity during the inference phase. This conclusion also opens up a new avenue for the further application of C3oT.

Conclusion

We introduce C3oT, a simple but effective method for CoT compression. Through comprehensive experiments and analyses, we demonstrate that our approach holds significant practical implications, as it enables models, which are trained using complex, longer CoT to enhance reasoning capabilities, to be applied in time-sensitive real-world applications.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aggarwal, S.; Mandowara, D.; Agrawal, V.; Khandelwal, D.; Singla, P.; and Garg, D. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3050–3065.

Amini, A.; Gabriel, S.; Lin, P.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Chase, H. 2022. LangChain. https://github.com/langchain-ai/langchain. Accessed: 2023-12-09.

Chevalier, A.; Wettig, A.; Ajith, A.; and Chen, D. 2023. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Das, R. J.; Ma, L.; and Shen, Z. 2023. Beyond size: How gradients shape pruning decisions in large language models. *arXiv preprint arXiv:2311.04902*.

Deng, Y.; Prasad, K.; Fernandez, R.; Smolensky, P.; Chaudhary, V.; and Shieber, S. 2023. Implicit Chain of Thought Reasoning via Knowledge Distillation. *arXiv preprint arXiv:2311.01460*.

Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332.

Frantar, E.; and Alistarh, D. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, 10323–10337. PMLR.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Ge, T.; Hu, J.; Wang, X.; Chen, S.-Q.; and Wei, F. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Jiang, H.; Wu, Q.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2023a. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Jiang, H.; Wu, Q.; Luo, X.; Li, D.; Lin, C.-Y.; Yang, Y.; and Qiu, L. 2023b. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Jin, M.; Yu, Q.; Zhao, H.; Hua, W.; Meng, Y.; Zhang, Y.; Du, M.; et al. 2024. The Impact of Reasoning Step Length on Large Language Models. *arXiv preprint arXiv:2401.04925*.

Kim, S.; Shen, S.; Thorsley, D.; Gholami, A.; Kwon, W.; Hassoun, J.; and Keutzer, K. 2022. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 784–794.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Lampinen, A. K.; Dasgupta, I.; Chan, S. C.; Matthewson, K.; Tessler, M. H.; Creswell, A.; McClelland, J. L.; Wang, J. X.; and Hill, F. 2022. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.

Li, Y.; Dong, B.; Lin, C.; and Guerin, F. 2023. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*.

Liu, T.; Chen, Z.; Liu, Z.; Tian, M.; and Luo, W. 2024. Expediting and Elevating Large Language Model Reasoning via Hidden Chain-of-Thought Decoding. *arXiv preprint arXiv:2409.08561*.

Maini, P.; Seto, S.; Bai, H.; Grangier, D.; Zhang, Y.; and Jaitly, N. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*.

Marasović, A.; Beltagy, I.; Downey, D.; and Peters, M. E. 2021. Few-shot self-rationalization with natural language prompts. *arXiv preprint arXiv:2111.08284*.

Merrill, W.; and Sabharwal, A. 2023. The Expresssive Power of Transformers with Chain of Thought. *arXiv preprint arXiv:2310.07923*.

Mitra, A.; Del Corro, L.; Mahajan, S.; Codas, A.; Simoes, C.; Agarwal, S.; Chen, X.; Razdaibiedina, A.; Jones, E.; Aggarwal, K.; et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Modarressi, A.; Mohebbi, H.; and Pilehvar, M. T. 2022. Adapler: Speeding up inference by adaptive length reduction. *arXiv preprint arXiv:2203.08991*.

Mu, J.; Li, X.; and Goodman, N. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

Mukherjee, S.; Mitra, A.; Jawahar, G.; Agarwal, S.; Palangi, H.; and Awadallah, A. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Nye, M.; Andreassen, A. J.; Gur-Ari, G.; Michalewski, H.; Austin, J.; Bieber, D.; Dohan, D.; Lewkowycz, A.; Bosma, M.; Luan, D.; et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Pan, Z.; Wu, Q.; Jiang, H.; Xia, M.; Luo, X.; Zhang, J.; Lin, Q.; Rühle, V.; Yang, Y.; Lin, C.-Y.; et al. 2024. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. *arXiv preprint arXiv:2403.12968*.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q. V.; Chi, E. H.; Zhou, D.; et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, G.; Cheng, S.; Zhan, X.; Li, X.; Song, S.; and Liu, Y. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Wingate, D.; Shoeybi, M.; and Sorensen, T. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. *arXiv preprint arXiv:2210.03162*.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 38087–38099. PMLR.

Zhang, L.; Zhang, Y.; Ren, K.; Li, D.; and Yang, Y. 2023. Mlcopilot: Unleashing the power of large language models in solving machine learning tasks. *arXiv preprint arXiv:2304.14979*.