Cascaded Chain-of-Thoughts Distillation: Distilling Reasoning Capabilities from Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown remarkable reasoning capabilities at increased scales, spurring efforts to distill such capabilities into smaller, compact models via teacher-student learning. Previous works directly fine-tune student models on teachers' generated Chain-of-Thoughts (CoTs) data or learn it in a multi-task framework. However, these methods struggle with CoTs generalization due to spurious correlations between questions and answers, as well as inconsistencies in the logic connecting the rationales to the answers. In this paper, we propose **Cas**caded **CoT**s **D**istillation (CasCoD), a straightforward but effective method to address these issues. Specifically, we decompose the full CoTs distillation into two comprehensive tasks and learn it in a cascade way by sharing the input prefix. By separating and cascading the tasks, CasCoD not only enables the student model to concentrate on reasoning without the distraction of answers but ensures faithful reasoning in students, thus enhancing the generalizability of CoTs. Extensive experiments and further analysis demonstrate the effectiveness of CasCoD on both in-domain and out-of-domain benchmark reasoning datasets.

1 Introduction

Recent developments in large language models (LLMs) have brought remarkable improvements in reasoning via Chain-of-Thought (CoT) prompting (Wei et al., 2022b). However, these great reasoning capabilities are often associated with more parameters (Wei et al., 2022a), which is not practical to emergent in smaller language models (SLMs). To address this, there is a lot of work (Magister et al., 2023; Ho et al., 2023; Shridhar et al., 2023; Fu et al., 2023) trying to make the reasoning capabilities isolated and distilled to SLMs by directly fine-tuning on teacher LLMs generated CoTs data. This process, known as standard CoTs distillation, requires SLMs to generate CoTs in a single step. However,

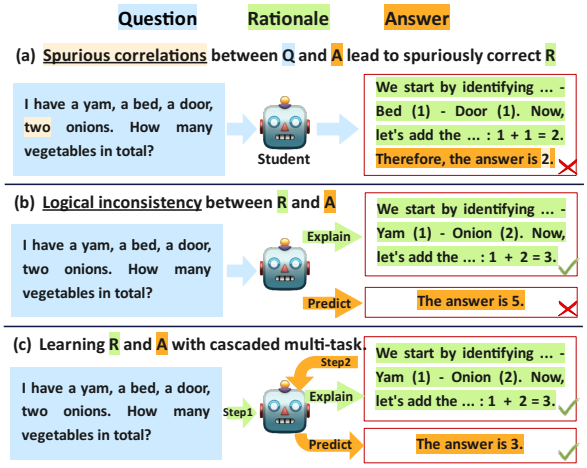

Figure 1: An illustration of the distinction between our approach and previous methods. (a) Standard CoTs distillation suffers from inefficient CoTs learning by the spurious correlation between the question and answer, leading to poor out-of-distribution generalization; (b) Multi-task distillation makes students inconsistently reason due to the isolated tasks; (c) Our approach decomposes CoTs and learn its in a cascaded multi-task way to address these issues.

due to the influence of spurious correlations between the question and the answer (Wang et al., 2023a), the student may learn reasoning shortcuts, which in turn reduces the generalizability of the intermediate reasoning process (rationale), as shown in Figure 1 (a). Some studies (Hsieh et al., 2023; Li et al., 2022) employ a multi-task learning framework to distill the CoTs by learning the rationale (the intermediate reasoning process) and the answer separately. However, this approach may result in logical inconsistency, with the student's rationale failing to support the correctness of its predicted answer, as illustrated in Figure 1 (b).

The flaws of the above two methods result in CoTs distilled student models that perform worse on unseen reasoning tasks than models directly fine-tuned on answers [1]. Different from recent works

[1]Confirmed experimentally, see §4.3.

(Wang et al., 2023c; Liu et al., 2023) that primarily focus on in-domain tasks, we argue that **the challenge of CoTs distillation lies in empowering students to truly learn reasoning—not only to excel in seen tasks but also to correctly reason for unseen problems**.

To address the above issues, we posit that the essence of effective CoTs distillation is tailoring the CoT learning approach for student models with limited parameters. This involves breaking down the CoT to enable students to learn in a manner that is both coherent and focused. Building on this basic insight, we propose **Cas**caded **C**hain-**o**f-Thoughts **D**istillation (CasCoD), a straightforward but effective CoTs distillation method that decomposes the full CoT learning into a series of successive, cascaded learning tasks. Specifically, we reorganize the standard CoTs distillation from Question to (Rationale, Answer) into two consecutive steps: first from Question to Rationale, and then from (Question, Rationale) to Answer, as shown in Figure 1 (c). In the first step, the student model is not required to consider the answer, allowing it to focus solely on the rationale learning, thereby enhancing the generalizability of the CoT. In the second step, where the student predicts the answer based on the question and rationale, it ensures that students can engage in faithful reasoning.

We conduct extensive experiments to assess the CoT reasoning capabilities of the distilled student model across both in-domain (IND) and out-of-domain (OOD) benchmark reasoning datasets. Experiments show that: (1) Previous CoTs distillation methods underperform in OOD tasks compared to direct fine-tuning on answers, but our proposed method CasCoD overcomes this limitation by learning the CoT in a cascade way. (2) CasCoD significantly outperforms the best distillation baselines on both IND and OOD tasks, achieving an average improvement of 6.4%. (3) CasCoD is universally applicable to student SLMs of varying sizes and outperforms the standard CoTs distillation with much less training data. (4) Different components within CoTs should be allocated different attention and learned across multiple steps rather than in a single step for better generalizability. (5) An additional faithfulness evaluation experiment demonstrates that student SLMs distilled by CasCoD can generate more self-consistent CoTs compared to the distillation baselines.

2 Related Works

CoT Capability of Language Models LLMs have demonstrated a wide array of capabilities in numerous Natural Language Processing (NLP) tasks, underscored by various studies (Chowdhery et al., 2023; Wei et al., 2022a). One notable manifestation of this is the Chain-of-Thought (CoT) prompting method (Wei et al., 2022b), which facilitates models in articulating a series of deductive reasoning steps. This method has substantially enhanced LLMs' problem-solving abilities, as evidenced in several works (Kojima et al., 2022a; Wang et al., 2023b; Huang et al., 2023). Despite these advancements, the effectiveness of CoT prompting notably diminishes in smaller models (Wei et al., 2022a). Research by Chung et al. (2022) indicates that with targeted training on CoT data via instruction tuning, SLMs can unlock CoT capabilities. In our study, we demonstrate that SLMs' CoT performance can be further enhanced by decomposing the complete CoT training process into a structured sequence of progressive learning tasks.

Distilling Knowledge from LLMs Numerous studies (Taori et al., 2023; Chiang et al., 2023; Peng et al., 2023) have explored the knowledge distillation from advanced, proprietary LLMs like ChatGPT (OpenAI, 2023), employing strategies akin to black-box model extraction (Krishna et al., 2020; Dai et al., 2023) or model imitation (Gudibande et al., 2023). These efforts typically concentrate on distilling a broad range of abilities via instruction tuning on extensive and varied datasets (Xu et al., 2023; Wu et al., 2023; Jiang et al., 2023). Our work, however, is aimed at distilling the CoT reasoning capabilities from LLMs, aligning with the objectives of Magister et al. (2023); Ho et al. (2023), who propose a standard CoTs distillation method that directly fine-tunes SLMs on CoTs produced by teacher LLMs. Fu et al. (2023) expands on this by fine-tuning with various reasoning data formats for specializing domain-specific SLMs. Additionally, Wang et al. (2023c) distill SLMs via learning from self-reflection and feedback in an interactive, multi-round paradigm with teacher LLMs. However, the above methods are derivatives of the standard CoT distillation which suffers from inefficient CoT learning by the spurious correlation. In contrast, we decompose the CoT distillation into multi-task distillation to enable students to focus on learning reasoning for better generalizability.

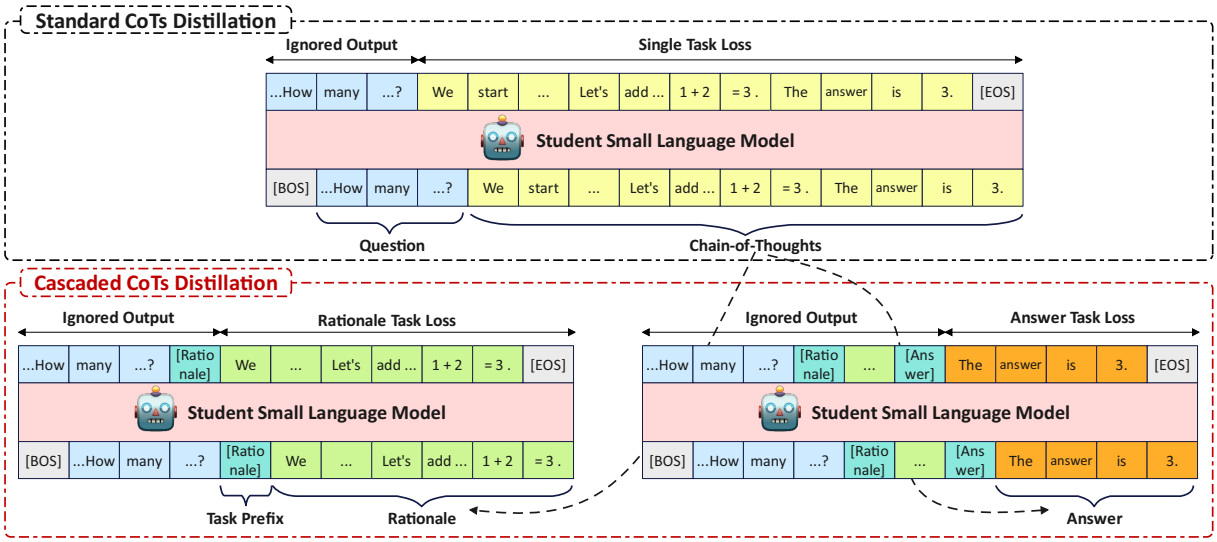

Figure 2: Overview of our proposed method **Cascaded CoTs Distillation (CasCoD)**. Different from the standard CoTs distillation, we decompose the CoTs into rationales and answers and learn them in a cascaded way by adding task prefixes and labels of pre-task into inputs for student models.

Multi-task CoTs Distillation Hsieh et al. (2023) propose to learn the rationale and answers by adding task prefixes to the input as separate goals for optimizing. Li et al. (2022) propose learning two tasks including the entire CoTs and the single answers to enhance the reasoning of student SLMs. Based on these foundations, Liu et al. (2023) introduce an additional distillation objective, self-evaluation, aiming for SLMs to assess the accuracy of their CoTs akin to LLMs' evaluative processes. Recognizing that previous methods fail to link multiple tasks, potentially confusing learners, we introduce cascaded multi-task learning to clarify the learning process and ensure faithful reasoning.

3 Methodology

We propose a novel distillation method that decomposes the original one-time CoT distillation training process into two consecutive training steps, as illustrated in Figure 2. We aim to tailor the learning process for the student model by dividing the training into cascaded, simpler stages. Formally, the standard CoTs distillation objective $\mathbf{q} \rightarrow \mathbf{r}, \mathbf{a}$ is split into two distinct learning tasks: $\mathbf{q} \rightarrow \mathbf{r}$ and $\mathbf{q}, \mathbf{r} \rightarrow \mathbf{a}$, each tagged with a unique task prefix. Below we describe the vanilla CoTs distillation method in §3.2 and then discuss the limitations and propose our method in §3.3.

3.1 Extract Rationale From Teacher LLMs

The initial phase of the distillation is to derive CoTs from teacher LLMs for each question-answer pair $\{q, a\}$ in a raw dataset. This involves using a CoT prompting technique (Wei et al., 2022b), detailed in Appendix C.1, which guides the teacher LLMs to generate CoTs that follow a prescribed format with multiple reasoning steps. The prompt template is shown in Appendix C.1. It's important to note that the rationale r and answers a produced by the LLMs may not always align with accuracy. To maintain CoT quality, we selectively retain only those that match the ground truth in the dataset, effectively building a CoT dataset $\mathcal{D} = \{q, r, a\}$ for training the student model.

3.2 Preliminaries for CoTs Distillation

The **Standard CoTs Distillation** (Magister et al., 2023; Ho et al., 2023), often referred to as single-task learning, is to teach SLMs to generate the rationale and answer in one time as follows:

$$\mathcal{L}_{\text{single}} = \mathbb{E}_{q,r,a\sim\mathcal{D}}\left[\ell\left(q, r \oplus a\right)\right] \quad (1)$$

where ℓ signifies the negative log-likelihood loss function, expressed as:

$$\ell\left(x, y\right) = -\sum_{y_t \in y} \log P\left(y_t \mid x, y_{<t}\right) \quad (2)$$

This approach differs from **Multi-task Learning** (Hsieh et al., 2023), where taking rationale generation as an auxiliary task besides answer prediction and training the two tasks in parallel as:

$$\mathcal{L}_{multi} = \mathbb{E}_{q,r,a\sim\mathcal{D}}\left[\ell\left(q \oplus \mathcal{A}, a\right) + \lambda\ell\left(q \oplus \mathcal{R}, r\right)\right] \quad (3)$$

where $\mathcal{A}$ and $\mathcal{R}$ denote the task prefixes "[Answer]" and "[Rationale]", respectively. The parameter λ adjusts the emphasis on rationale generation loss.

3.3 Cascaded CoTs Distillation

As previously noted, the above two methods can lead to challenges in effectively learning rationales due to the spurious correlation between questions and answers, or they may cause students to neglect the logical consistency between rationales and answers, impacting the generalizability of CoTs. Our proposed CasCoD leverages the advantages of both methods and addresses their shortcomings by decomposing the CoT learning objective into two distinct but cascaded tasks: Rationale Learning and Answer Learning as shown in Figure 2.

Rationale Learning For rationale learning, each question q is combined with the task prefix $\mathcal{R}$ as the input for the student model, with the rationale r produced by the teacher serving as the label. The loss function of rationale learning is as follows:

$$\mathcal{L}_{rationale} = \mathbb{E}_{q,r,a\sim\mathcal{D}}\left[\ell\left(q \oplus \mathcal{R}, r\right)\right] \quad (4)$$

Answer Learning For answer learning, the input of student models is composed of the question q and the teacher's rationale r that is used as the label in the rationale learning, along with the task prefix A and the answer a serves as the label. The loss function of answer learning is thus:

$$\mathcal{L}_{answer} = \mathbb{E}_{q,r,a\sim\mathcal{D}}\left[\ell\left(q \oplus \mathcal{R} \oplus r \oplus \mathcal{A}, a\right)\right] \quad (5)$$

Weighted Multi-task Learning Loss To maintain emphasis in cascaded multi-task learning, we add adjustable weights α for each loss. The combined loss for CasCoD is given by:

$$\mathcal{L}_{cascaded} = (1-\alpha)\mathcal{L}_{rationale} + \alpha\mathcal{L}_{answer} \quad (6)$$

During inference, student models perform two forward computations, mirroring the training process: the first for generating rationales and the second for predicting the final answer.

Relationship with Standard CoTs Distillation It's important to note that in cascaded multi-task learning, the tasks share the same prefix, and the label of the previous task serves as the input for the next. Under the teacher-forcing training mode (Goodfellow et al., 2016), this might suggest at first glance that cascaded CoTs distillation closely resembles standard CoTs distillation, with the primary distinction being the introduction of weights for more nuanced, token-level learning adjustments. However, this perception overlooks a critical difference: CasCoD strategically segments the learning process, first focusing on rationale generation before proceeding to answer generation, a capability not achievable with standard CoTs distillation.

How about aligning standard CoTs distillation to cascade multi-task learning by adding special tokens such as task prefixes $\mathcal{R}, \mathcal{A}$ and end-of-sequence tokens $[\mathbf{EOS}]$ as follows?

$$\mathcal{L}_{\text{aligned}} = \mathbb{E}_{q,r,a\sim\mathcal{D}} \left[\omega\ell\left(q \oplus \mathcal{R}, r \oplus [\mathbf{EOS}] \oplus \mathcal{A} \oplus a\right)\right] \quad (7)$$

where ω denotes the token-level weights and $\mathcal{A}$ in the label can be masked in calculating the loss.

However, it turns out that this alignment cannot be completely achieved. This is because even if there is formal alignment, inserting an $[\mathbf{EOS}]$ token between rationale and answer means that during answer prediction, the attention mechanism (Vaswani et al., 2017) will make the student model notice this ending token. As a result, the student model may consider the subsequent generation task as an entirely new task unrelated to the preceding tasks, cutting off the connection between answer generation and the previous question and rationale. In §4.4, we will compare the effects of such "complete" alignment approach, the method utilizing only weight alignment, and our CasCoD approach.

4 Experiments

In this section, we conduct extensive experiments and comprehensive analysis to evaluate the effectiveness of our method across both in-domain (IND) and out-of-domain (OOD) datasets.

4.1 Datasets

4.1.1 In-domain

BIG-Bench Hard (BBH) (Suzgun et al., 2023) comprises 27 challenging tasks covering arithmetic, symbolic reasoning et al. from BIG-Bench (BB) (Guo et al., 2023). The majority of the data involve multiple-choice questions, with a few being open-ended. To underscore the superiority of our approach, we chose to perform distillation on this most challenging dataset. Specifically, we randomly divide the BBH dataset into a training set (BBH-train) for distillation and a test set (BBH-test) as the IND evaluation task, in a 4:1 ratio.

4.1.2 Out-of-domain

BIG-Bench Sub (BB-sub). BB is a popular benchmark consisting of 203 tasks covering a wide range of topics, including mathematics, commonsense reasoning, and various other domains. For ease of evaluation, we filter the subtasks within BB based on subtask keywords, specifically focusing on tasks related to "multiple-choice" and "reasoning"[2], and ensure that tasks from BBH were excluded, resulting in 61 subtasks. Then we randomly sample up to 100 instances for each subtask, resulting in the creation of BB-sub.

AGIEval (Zhong et al., 2023) is a renowned human-centric benchmark used to assess LMs' reasoning abilities, whose tasks span various domains, including college entrance exams (English / Math / Law), logic tests et al. We evaluate our method on the subtasks that are related to multiple-choice questions in the English language.

AI2 Reasoning Challenge (ARC) (Clark et al., 2018) consists of ARC-Easy (ARC-E) and ARC-Challenge (ARC-C). The distinction lies in ARC-E consisting of relatively simpler questions from middle and high school science exams, while ARC-C comprises more complex and challenging questions. We utilize the testing set of the ARC dataset for evaluation. The statistics of the all above datasets can be found in Appendix B.1.

4.2 Models & Baselines & Setup

Models We employ the contemporary, popular open-source language model LLaMA2-7B (Touvron et al., 2023) as the student SLM. Considering the pricing and capabilities, we utilize OpenAI's powerful black-box LLM, ChatGPT[3], as the teacher. We query ChatGPT to annotate the CoT data with the same manual prompt used in the previous work (Suzgun et al., 2023).

Baselines We compare our method with the following baselines: (1) **Teacher & Vanilla Student** under various settings, e.g., Zero-shot (+CoT) or Few-shot (+CoT), for showing the impact of distilling reasoning ability from LLMs. (2) **Std-CoT** (Magister et al., 2023), which is a standard CoTs distillation method that directly fine-tune student models on the CoTs data. (3) **Step-by-step** (Hsieh et al., 2023) is a multi-task CoT distillation method that separately optimizes the objectives of answer learning and inference process learning. (4) **MT-CoT** (Li et al., 2022) is also a multi-task CoTs distillation method, but unlike Step-by-step, it simultaneously optimizes the objectives of answer prediction and CoTs learning. (5) **SCOTT** that enhances the reasoning consistency of the student model by introducing additional counterfactual data.

Setup We employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning of the student SLMs. We empirically set α in multi-task learning as 0.3. All experiments are conducted using a mixed-precision training strategy on 4 $\times$ A100 GPUs. For the inference stage, vLLM[4] (Kwon et al., 2023) is utilized to accelerate inference, employing a greedy decoding strategy to generate text on one single A100 GPU. Further details on training and hyper-parameters can be found in Appendix B.2.

4.3 Main Results

Table 1 presents the automatic evaluation results of our proposed CasCoD and baselines on in-domain (IND) and out-of-domain (OOD) datasets.

CoTs distillation enhances the reasoning performance of students. Comparing with the Zero-shot-CoT and Few-shot-CoT settings of student models, the performance of those with distillation is significantly improved by learning the teacher LLM's CoTs. Except for BB-sub, the student model has 3-4 times improvement compared to vanilla ones across all datasets.

CasCoD overcomes limitations of distillation baselines in OOD performance. From the table, we can find that Answer-SFT on the OOD datasets outperforms all the distillation baselines by an average of 5%, which indicates that it seems student models' performance decreases when learning the CoTs. This pattern is also noticeable in models without distillation, as evidenced by the comparison between Zero-shot and Zero-shot-CoT (or Few-shot and Few-shot-CoT) settings. We attribute this to spurious correlations between questions and answers in these implicit reasoning task datasets (Gururangan et al., 2018; Zellers et al., 2019; Blodgett et al., 2020), which students can easily learn by directly fine-tuning. The distillation baselines that require students to consider predicting answers

[2]For detailed descriptions of the subtasks in BIG-Bench, please refer to https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md.

[3]https://chat.openai.com/chat. We utilize the $gpt-3.5-turbo-0613$ for CoTs extraction.

[4]https://github.com/vllm-project/vllm

Method	Distill?	Gen CoT?	BBH-test	BB-sub	AGIEval	ARC-E	ARC-C	AVG
In-domain?			✓	×	×	×	×	
Teacher: ChatGPT (gpt-3.5-turbo)								
Zero-shot-CoT	×	✓	42.7	44.1	49.5	91.9	81.1	61.9
Few-shot-CoT	×	✓	73.1	-	-	-	-	-
Student: LLaMA2-7B								
Zero-shot	×	×	14.8	15.5	6.9	18.2	13.9	13.9
Zero-shot-CoT	×	✓	10.6	7.7	7.1	18.4	14.8	11.7
Few-shot	×	×	15.1	28.5	25.5	25.5	25.4	24.0
Few-shot-CoT	×	✓	16.3	25.3	9.9	17.2	17.2	17.2
Answer-SFT	×	×	51.5	33.2	31.2	71.6	53.7	48.2
Std-CoT (Magister et al., 2023)	✓	✓	54.2	28.7	21.6	59.6	45.1	41.8
SCOTT (Wang et al., 2023a)	✓	✓	42.4	18.8	13.0	45.7	34.1	30.8
MT-CoT (Li et al., 2022)	✓	✓	56.8	30.3	22.0	49.4	38.2	39.3
Step-by-step (Wang et al., 2023c)	✓	✓	42.4	27.7	**28.8**	68.5	48.6	43.2
CasCoD (ours)	✓	✓	**59.4**$_{+2.6}$	**37.0**$_{+6.7}$	28.3$_{-0.5}$	**70.6**$_{+2.1}$	**52.7**$_{+3.9}$	**49.6**$_{+6.4}$

Table 1: Accuracy (%) on in-domain and out-of-domain datasets with different methods. We employ "Let's think step by step" (Kojima et al., 2022b) for Zero-shot-CoT settings and the manually curated prompt (Suzgun et al., 2023) for Few-shot-CoT settings. The best performance among distilled student models is marked in **bold**. The subscript shows the performance gap between our method and the best baselines on each dataset.

while generating the rationale, inadvertently make the simpler task of answer prediction interfere with the rationale learning, thus reducing the generalization of CoTs. In contrast, CasCoD not only surpasses Answer-SFT by 7.9% in IND datasets but also achieves comparable results in OOD scenarios. This underscores the effectiveness of our strategy, which involves decomposing CoTs and engaging in cascaded multi-task learning, in enhancing reasoning capabilities across diverse datasets.

CasCoD significantly outperforms the distillation baselines across IND and OOD datasets. From Table 1, it can be observed that CasCoD outperforms baselines on both IND and OOD datasets in most cases. Specifically, CasCoD secures an average in-domain improvement of 5.2% and an out-of-domain enhancement of 8.4% over the Std-CoT, along with an overall 6.4% improvement compared to the multi-task learning (Step-by-step) approach. Impressively, CasCoD achieves 80.1% of the teacher LLM's performance in Zero-shot-CoT settings. These results underscore the efficacy of CasCoD, significantly boosting the generative capabilities of CoTs on unseen tasks.

4.4 Ablation Study

CasCoD is universally applicable to models of varying sizes. We perform model distillation on TinyLLaMA-1.1B[5] (Zhang et al., 2024), LLaMA2-7B, and LLaMA2-13B, respectively and compare with standard CoTs distillation (Std-CoT) and multi-task distillation (MT-CoT & Step-by-step). In Figure 3 and 7, we can find that CasCoD consistently outperforms the baselines on both IND and OOD datasets across various sizes of student models. Notably, the performance improvement of our method is the most obvious in the BB-sub, where the performance of the 13B student model reaches 92.7% of the teacher LLM's performance. Furthermore, as model sizes increase, the performance gap between CasCoD and the baselines widens on OOD datasets, highlighting CasCoD's superior efficiency in distilling CoTs for larger models.

CasCoD significantly outperforms standard CoTs distillation on OOD with much less training data. In Figure 4, CasCoD achieves a 6.3% improvement over Std-CoT on the BB-sub dataset, using only 25% of the full BBH-train data. In the case of other OOD datasets, CasCoD requires merely 12.5% of the full training data to surpass the Std-CoT trained with the full dataset by 5% to 7% in performance. These results demonstrate the efficiency of CasCoD, capable of enhancing CoTs generalization with a smaller amount of CoTs data.

Rationales and answers should be allocated varying levels of attention and learned across

[5] https://huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T

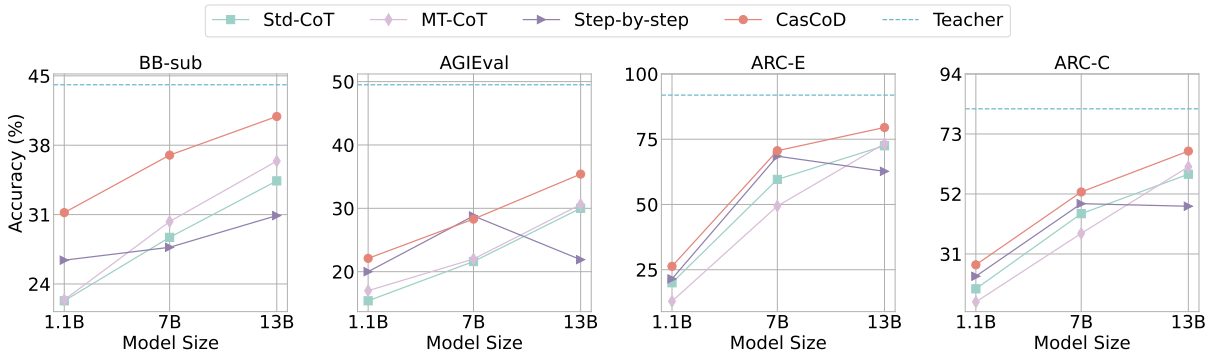

Figure 3: Ablation study on model size for four OOD datasets. The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting. The results in IND dataset can be found in Appendix A.1.

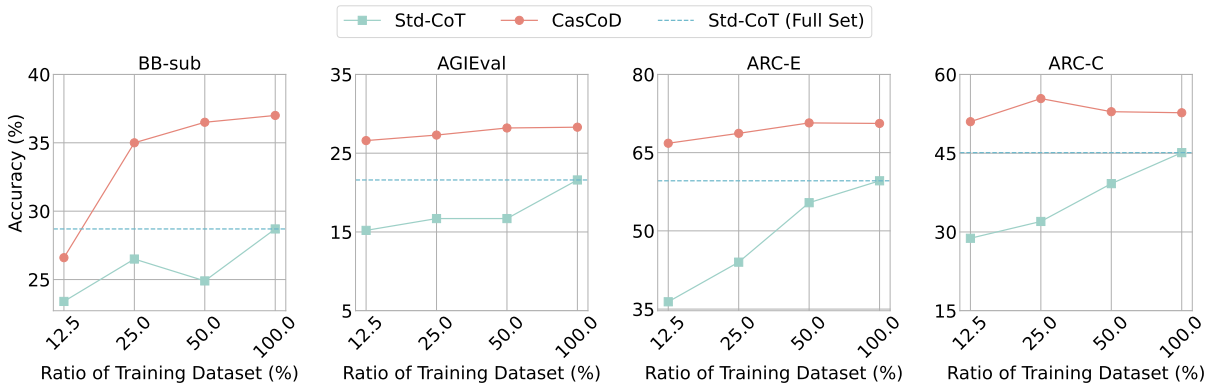

Figure 4: Ablation study on training data size for four OOD datasets. The dotted line indicates the performance of fine-tuning the student models by standard CoTs distillation using the full set (100% of) BBH-train dataset. The results in IND dataset can be found in Appendix A.2.

multiple steps rather than in a single step. As mentioned in §3.3, we further experimentally explore the relationship between CasCoD and single-task learning (e.g. Std-CoT) under the teacher forcing. In Figure 5, we can see that the "Weight Aligned" method, which simply adds weights compared to Std-CoT, enhances performance on both IND and OOD datasets, highlighting the benefit of tailored attention levels for rationales and answers. Moreover, the "All Aligned" shows significant improvement in OOD datasets compared to the "Weight Aligned" by incorporating task prefixes and sentence-ending tokens, suggesting that separating the learning phases for rationales and answers helps in minimizing distractions. However, we also notice that even attempting to align with CasCoD in single-step, there remains a performance gap. This suggests that despite using teacher forcing mode, single-task learning cannot fully align with multi-task learning, as the introduction of [EOS] tokens disrupts the correlation between multiple tasks.

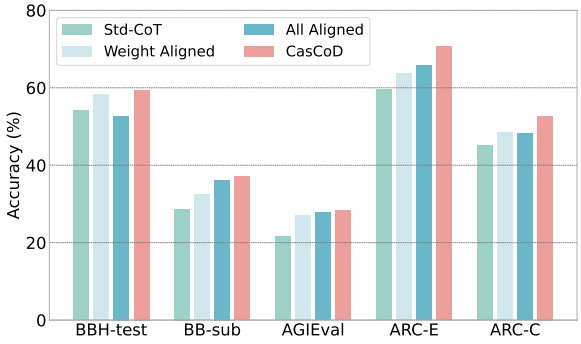

Figure 5: Ablation study on multi-steps across the IND and OOD datasets. "Weight Aligned" refers to the approach of adding token-level weights in the Std-CoT. "All Aligned" builds upon "Weight Aligned" by further incorporating special task prefixes and end-of-sequence tokens, aiming for structural alignment with CasCoD. For both single step settings, we set weights as 0.3 on the answer tokens and 0.7 on the rationale tokens.

4.5 Robustness W.R.T. Weights

In this subsection, we explore how variations in weights affect the performance of models with

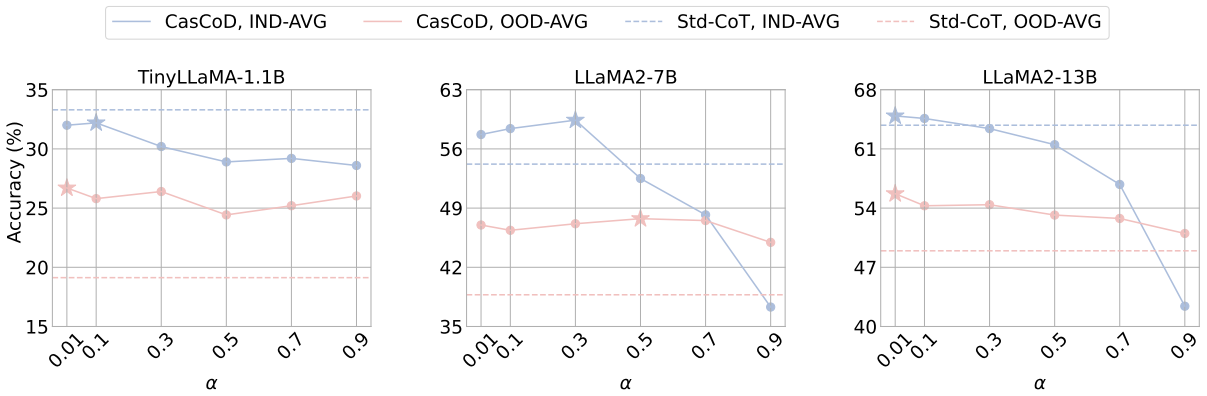

Figure 6: Ablation study on task weights α. The results are reported by IND-AVG and OOD-AVG that respectively denote averge accuracy on IND and OOD datasets. The best performance among weights are marked with "☆".

different parameter sizes on both IND and OOD datasets, as shown in Figure 6.

Students' performance is not sensitive to weights on OOD datasets. From the figure, we observe that regardless of weight changes, CasCoD consistently outperforms Std-CoT in OOD by average, even at $\alpha = 0.9$ (meaning the model allocated only 10% of its attention to rationales generation). This demonstrates that CasCoD exhibits robust generalization in OOD and also underscores the effectiveness of decomposing CoTs for distillation.

CasCoD is more robust for smaller student models. We observe that the 1.1B model shows less variation in performance compared to the 7B and 13B models in IND. Notably, the performance of the 13B model drops sharply as α changes from 0.5 to 0.9, indicating that larger models are more susceptible to weight adjustments in the IND dataset.

Prioritizing the rationale over the answer yields better results. It is evident that across different model sizes, the optimal weights on both IND and OOD datasets are less than 0.5, which indicates that focusing on the rationale enables student models to learn CoTs with greater generalizability.

4.6 Faithfulness of Students

To ensure that the rationale provided by students supports their predicted answers, another metric for evaluating CoTs distillation is the faithfulness of students. Following the previous work (Wang et al., 2023a), we use the LAS metric (Hase et al., 2020), whose core idea is to measure the extent that the rationales r' aid a simulator in predicting the answers a', defined as:

$$LAS = Acc\left(q, r' \rightarrow a'\right) - Acc\left(q \rightarrow a'\right) \quad (8)$$

where we employ ChatGPT and GPT4 as the simulator, respectively. The results are shown in Table 2. CasCoD is observed to generate rationales that are more consistent than those from baselines, particularly MT-CoT and step-by-step methods. This suggests that despite CasCoD being a multi-task learning framework, the introduction of cascading learning ensures that students can faithfully reason.

Method	ChatGPT	GPT4	AVG
Std-CoT	40.8	29.8	35.3
SCOTT	36.2	29.4	32.8
MT-CoT	36.2	25.8	31
Step-by-step	6.6	-0.1	3.25
CasCoD (ours)	**40.8**	**31.6**	**36.2**

Table 2: Faithfulness (LAS, %) of the compared methods with different LLM evaluators on the IND dataset. The prompt templates can be found in Appendix C.2

.

5 Conclusion

This work presents a staightforward yet effective CoTs distillation method CasCoD as a solution to the challenges of distilling reasoning capabilities from LLMs into smaller ones. Specifically, we break down the full CoT distillation process into two cascade tasks by sharing the input prefix, leading to enhanced CoTs generalizability. Extensive experiments show that our proposed method significantly outpeforms the baselines across both in-domain and out-of-domain benchmark reasoning datasets. Further analysis reveals that CasCoD is robust to model size and task weights and can lead to a faithful student models.

Limitations

Considering the cost such as API calls and GPU training expenses, we only choose ChatGPT as the teacher LLM and the widely-used model LLaMA2 as the student SLM. Employing GPT-4 as the teacher provides high-quality CoTs, which could better validate the effectiveness of our proposed method CasCoD. Besides, when distilling the student model using CasCoD, it requires two forward computations, increasing the training time cost. Additionally, some research (Schaeffer et al., 2023a) indicates that the emergent abilities of LLMs are the result of accomplishing multiple sub-tasks correctly at the same time, leading to a quantitative change that results in a qualitative transformation. This paper merely explores distilling CoT into two steps, envisioning that CoT could be broken down into even more steps to allow the student model to focus on learning specific tasks. However, the questions of how to decompose CoTs, when to do so, and how to allocate attention remain unresolved. We leave these issues for future research.

Ethics Statement

It is important to note that this work utilizes CoT data extracted from ChatGPT for distillation, which may result in inheriting the social biases (Schaeffer et al., 2023b) and hallucination (Zhang et al., 2023) present in LLMs. However, we are optimistic that future advancements in resolving these issues in LLMs will naturally lead to the development of student models with reduced toxicity.

References

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna M. Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *ACL*, pages 5454–5476. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Chengwei Dai, Minxuan Lv, Kun Li, and Wei Zhou. 2023. Meaeq: Mount model extraction attacks with efficient queries. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12671–12684. Association for Computational Linguistics.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *CoRR*, abs/2305.15717.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Beyond imitation: Leveraging fine-grained quality signals for alignment. *CoRR*, abs/2311.04072.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A.

Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT (2)*, pages 107–112. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *EMNLP (Findings)*, volume EMNLP 2020 of *Findings of ACL*, pages 4351–4367. Association for Computational Linguistics.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *ACL (1)*, pages 14852–14882. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *ACL (Findings)*, pages 8003–8017. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *EMNLP*, pages 1051–1068. Association for Computational Linguistics.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3134–3154. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022a. Large language models are zero-shot reasoners. In *NeurIPS*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022b. Large language models are zero-shot reasoners. In *NeurIPS*.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! model extraction of bert-based apis. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pages 611–626. ACM.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better. *CoRR*, abs/2210.06726.

Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. 2023. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models. *CoRR*, abs/2311.09214.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adámek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1773–1781. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt (June 13 version). https://chat.openai.com.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023a. Are emergent abilities of large language models a mirage? *CoRR*, abs/2304.15004.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023b. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *ACL (Findings)*, pages 7059–7073. Association for Computational Linguistics.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura,

Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023a. SCOTT: self-consistent chain-of-thought distillation. In *ACL (1)*, pages 5546–5558. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *ICLR*. OpenReview.net.

Zhaoyang Wang, Shaohan Huang, Yuxuan Liu, Jiahai Wang, Minghui Song, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023c. Democratizing reasoning ability: Tailored learning from large language model. In *EMNLP*, pages 1948–1966. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *CoRR*, abs/2304.12244.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL (1)*, pages 4791–4800. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball. *CoRR*, abs/2305.13534.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *CoRR*, abs/2401.02385.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *CoRR*, abs/2304.06364.

A Ablation Study on In-domain Dataset

A.1 W.R.T. Model Size

The results of the model size ablation study on IND datasets are presented in Figure 7. We observe that CasCoD outperforms the baselines on both the 7B and 13B model and significantly surpasses the teacher LLMs in the Zero-shot CoT setting.

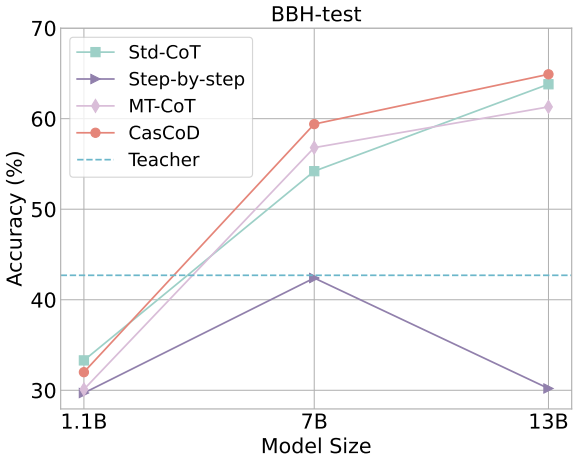

Figure 7: Ablation study on model size in the IND (BBH-test). The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting.

A.2 W.R.T. Training Data Size

The results of the training data ablation study on IND datasets, as shown in Figure 8, indicate that CasCoD outperforms standard CoT distillation across various sizes of training data. This demonstrates the efficiency of our proposed method.

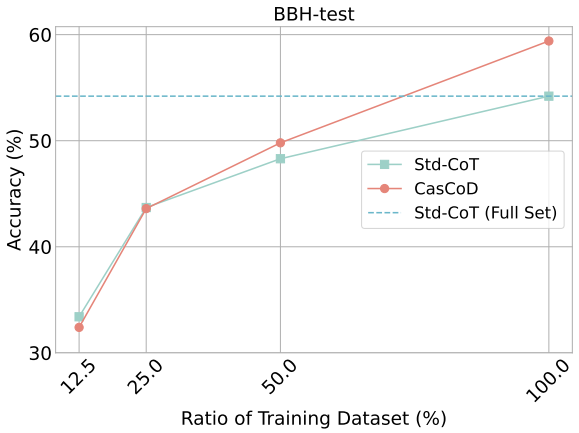

Figure 8: Ablation study on training data size in the IND (BBH-test). The dotted line indicates the performance of fine-tuning the student models by standard CoT distillation using the full set (100% of) BBH-train dataset.

B Details of Experiment

B.1 Dataset Statistics

Table 3, Table 4, Table 5 and Table 6 show the data statistics of AGIEval, ARC, BIG-Bench Hard (BBH) and BIG-Bench Sub (BB-sub), respectively.

No.	Task	Size	# Choices
1	AQuA-RAT	254	5
2	LogiQA-EN	651	4
3	LSAT-AR	230	5
4	LSAT-LR	510	5
5	LSAT-RC	269	5
6	SAT-Math	220	4
7	SAT-EN	206	4
8	SAT-EN (w/o Psg.)	206	4
	Sum	2546	-

Table 3: Statistics of AGIEval dataset.

Task	Size	# Choices
ARC-E	2376	4-5
ARC-C	1172	4-5

Table 4: Statistics of ARC test dataset.

No.	Task	Size	# Choices
1	Boolean Expressions	250	2
2	Causal Judgement	187	2
3	Date Understanding	250	6
4	Disambiguation QA	250	4
5	Dyck Languages	250	-
6	Formal Fallacies Syllogisms Negation	250	2
7	Geometric Shapes	250	11
8	Hyperbaton (Adjective Ordering)	250	2
9	Logical Deduction (3 objects)	250	3
10	Logical Deduction (5 objects)	250	5
11	Logical Deduction (7 objects)	250	7
12	Movie Recommendation	250	5
13	Multi-Step Arithmetic	250	-
14	Navigate	250	2
15	Object Counting	250	-
16	Penguins in a Table	146	5
17	Reasoning about Colored Objects	250	18
18	Ruin Names	250	11
19	Salient Translation Error Detection	250	6
20	Snarks	178	2
21	Sports Understanding	250	2
22	Temporal Sequences	250	4
23	Tracking Shuffled Objects (3 objects)	250	3
24	Tracking Shuffled Objects (5 objects)	250	5
25	Tracking Shuffled Objects (7 objects)	250	7
26	Web of Lies	250	2
27	Word Sorting	250	-
	Sum	6511	-

Table 5: Statistics of BIG-Bench Hard dataset.

B.2 Hyperparameters Settings

The hyperparameters in training and inference can be found in Table 7 and Table 8, respectively.

C Prompts

C.1 Prompts of Generating CoTs for ChatGPT

We use the prompt template shown in Table 9 to call the ChatGPT API to generate the CoTs for the BBH-train datasets.

C.2 Prompts of Simulators

We use the prompt templates shown in Table 10 and Table 11 to call the ChatGPT and GPT4 API to predict the answers given a question or with an additional rationale, respectively.

No.	Task	Size	# Choices
1	abstract_narrative_understanding	100	5
2	anachronisms	100	2
3	analogical_similarity	100	7
4	analytic_entailment	70	2
5	cause_and_effect	100	2
6	checkmate_in_one	100	26
7	cifar10_classification	100	10
8	code_line_description	60	4
9	conceptual_combinations	100	4
10	crass_ai	44	4
11	elementary_math_qa	100	5
12	emoji_movie	100	5
13	empirical_judgments	99	3
14	english_russian_proverbs	80	4
15	entailed_polarity	100	2
16	entailed_polarity_hindi	100	2
17	epistemic_reasoning	100	2
18	evaluating_information_essentiality	68	5
19	fantasy_reasoning	100	2
20	figure_of_speech_detection	59	10
21	goal_step_wikihow	100	4
22	gre_reading_comprehension	31	5
23	human_organs_senses	42	4
24	identify_math_theorems	53	4
25	identify_odd_metaphor	47	5
26	implicatures	100	2
27	implicit_relations	82	25
28	indic_cause_and_effect	100	2
29	intersect_geometry	100	26
30	kanji_ascii	100	5
31	kannada	100	4
32	key_value_maps	100	2
33	logic_grid_puzzle	100	3
34	logical_args	32	5
35	logical_fallacy_detection	100	2
36	metaphor_boolean	100	2
37	metaphor_understanding	100	4
38	minute_mysteries_qa	100	4
39	mnist_ascii	100	10
40	moral_permissibility	100	2
41	movie_dialog_same_or_different	100	2
42	nonsense_words_grammar	50	4
43	odd_one_out	86	5
44	parsinlu_qa	100	4
45	physical_intuition	81	4
46	play_dialog_same_or_different	100	2
47	presuppositions_as_nli	100	3
48	riddle_sense	49	5
49	similarities_abstraction	76	4
50	simple_ethical_questions	100	4
51	social_iqa	100	3
52	strange_stories	100	2
53	strategyqa	100	2
54	swahili_english_proverbs	100	4
55	swedish_to_german_proverbs	72	4
56	symbol_interpretation	100	5
57	timedial	100	3
58	undo_permutation	100	5
59	unit_interpretation	100	5
60	vitaminc_fact_verification	100	3
61	winowhy	100	2
	Sum	5384	-

Table 6: Statistics of BIG-Bench sub dataset. We filter the original dataset by retrieving tasks with keywords "multiple choice" and randomly sample up to 100 examples per task. Note, the task in BBH will not be involved in BB-sub.

Hyperparameter	TinyLLaMA-1.1B	LLaMA2-7B	LLaMA2-13B
gradient accumulation steps	4	4	8
per device batch size	16	16	8
learning rate	2e-4	2e-4	2e-4
epoches	20	15	10
max length	1024	1024	1024
β of AdamW	(0.9,0.999)	(0.9,0.999)	(0.9,0.999)
ϵ of AdamW	1e-8	1e-8	1e-8
γ of Scheduler	0.95	0.95	0.95
weight decay	0	0	0
warmup ratio	0	0	0
rank of LoRA	64	64	64
α of LoRA	32	32	32
target modules	q_proj, v_proj	q_proj, v_proj	q_proj, v_proj
drop out of LoRA	0.05	0.05	0.05

Table 7: Training hyperparameters.

Arguments	Student	Teacher
do sample	False	True
temperature	-	0.2
top-p	1.0	1.0
top-k	-	-
max new tokens	1024	2048
# return sequences	1	1

Table 8: Generation configs of students and teachers.

```
{Task Description}. Your response should conclude with the format "Therefore, the answer is".

Q: {Task Example Question No.1}
A: Let's think step by step. {Human-Curated-CoTs}.

Q: {Task Example Question No.2}
A: Let's think step by step. {Human-Curated-CoTs}.

Q: {Task Example Question No.2}
A: Let's think step by step. {Human-Curated-CoTs}.

Q: {QUESTION}
A: Let's think step by step.
```

Table 9: Prompt template of gpt-3.5-turbo for generating the CoTs data.

system content	`You are a helpful and precise assistant for following the given instruction.`

user content	`[Instruction] {Please read the question and then give your answer based on the question without any explanations.}`
	
	`Task Description: {TASK_DESCRIPTION}`
	
	`Question: {QUESTION}`
	
	`Your Answer:`

Table 10: Prompt template of simulators for predicting the answers when given the question.

system content	`You are a helpful and precise assistant for following the given instruction.`

user content	`[Instruction] {Please read the question and the rationale, and then give your answer based on the question and the rationale without any explanations.}`
	
	`Task Description: {TASK_DESCRIPTION}`
	
	`Question: {QUESTION}`
	
	`Rationale: {RATIONALE}`
	
	`Your Answer:`

Table 11: Prompt template of simulators for predicting the answers when given the question and rationale.