Improve Student's Reasoning Generalizability through Cascading Decomposed CoTs Distillation

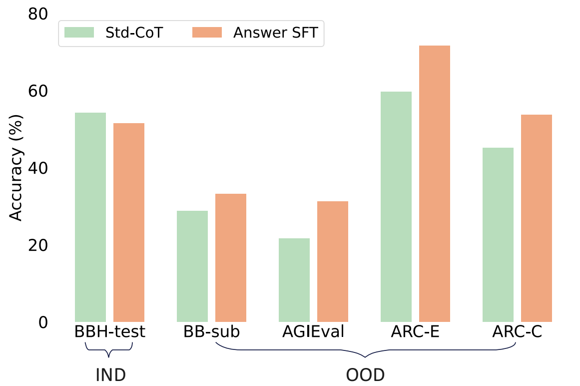
Chengwei Dai[1,2], **Kun Li**[1*], **Wei Zhou**[1], **Songlin Hu**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{daichengwei, likun2, zhouwei, husonglin}@iie.ac.cn

Abstract

Large language models (LLMs) exhibit enhanced reasoning at larger scales, driving efforts to distill these capabilities into smaller models via teacher-student learning. Previous works simply fine-tune student models on teachers' generated Chain-of-Thoughts (CoTs) data. Although these methods enhance in-domain (IND) reasoning performance, they struggle to generalize to out-of-domain (OOD) tasks. We believe that the widespread spurious correlations between questions and answers may lead the model to preset a specific answer which restricts the diversity and generalizability of its reasoning process. In this paper, we propose **Cas**cading Decomposed **Co**Ts **D**istillation (CasCoD) to address these issues by decomposing the traditional single-step learning process into two cascaded learning steps. Specifically, by restructuring the training objectives—removing the answer from outputs and concatenating the question with the rationale as input—CasCoD's two-step learning process ensures that students focus on learning rationales without interference from the preset answers, thus improving reasoning generalizability. Extensive experiments demonstrate the effectiveness of CasCoD on both IND and OOD benchmark reasoning datasets[1].

1 Introduction

Recent developments in LLMs have brought remarkable improvements in reasoning tasks via 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). Existing works (Magister et al., 2023; Ho et al., 2023; Fu et al., 2023; Zhou and Ai, 2024) try to make the reasoning capabilities isolated and distilled to student SLMs by simply fine-tuning on teacher LLMs generated CoTs data, known as standard CoTs distillation (Std-CoT). Although the method effectively leverages the LLMs' CoTs to boost the reasoning performance of student models on seen tasks, it does not ensure effective reasoning in OOD settings, leading to weak generalization on unseen tasks. Our pioneer study demonstrates that, as shown in Figure 1 (a), when using the same IND training dataset, student models developed via the method Std-CoT perform better on IND tasks but significantly worse on OOD tasks compared



(a) Answer SFT consistently outperform Std-CoT on OOD tasks.

Question: Why did someone bring a swimsuit to a ski resort?
Options:
(A) To swim in a heated pool.
(B) To wear as an underlayer for warmth.
(C) To use as a fashion statement.
(D) To participate in a polar bear plunge event.

Answer: (A) To swim in a heated pool.

(b) A case of spurious correlation between questions and answers.

Figure 1: (a) Empirical results of standard CoT distillation (Std-CoT) and directly fine-tuning on answer labels without CoTs (Answer SFT) on one in-domain (BBH-test) and the other four out-of-domain benchmark reasoning datasets. (b) In the given example, the semantic similarity between "swimsuit" in the question and "swim" in the answer demonstrates a high level of match, which could allow the model to predict the answer using simple keyword matching or certain rules.

*Kun Li is the corresponding author.
[1]Code available at https://github.com/C-W-D/CasCoD

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15623–15643
November 12-16, 2024

to models fine-tuned directly with question-answer pairs. The surprising findings indicate that students' CoTs do not effectively transfer to new domains and these SLMs seem to be more adept at learning to predict answers directly from questions.

We attribute these issues to the spurious correlations between questions and answers that are commonly found in implicit reasoning tasks (Gururangan et al., 2018; Zellers et al., 2019; Blodgett et al., 2020), as illustrated in Figure 1 (b). The Std-CoT approach requires models to learn both the rationale and the answer in a single step, where the learned spurious correlations in training stage can adversely affect the quality of rationale generation during inference. That is to say, upon reading a question, student models may fastly, unconsciously, and automatically formulate a "preset answer" (Hagendorff et al., 2022), which in turn may lead them to implicitly reduce the token generation space when producing CoT. This results in diminished diversity and generalizability of their rationales.

In this paper, we propose **Cas**cading decomposed **Co**Ts **D**istillation (CasCoD), a straightforward yet effective method to address these issues. Specifically, we decompose the traditional single-step learning process of Std-CoT into two cascaded learning steps: a rationale learning step and an answer learning step. In the rationale learning step, the training objective, with the answer removed, is defined as [2]: $q \rightarrow r$. In the answer learning step, we concatenate the question with the target output from the rationale learning step and use this combined input for the answer learning step, setting the training objective as $q, r \rightarrow a$. This cascading two-step learning configuration mitigates the capture of spurious correlations between questions and answers during the training phase, ensuring that students focus on learning rationales without interference from the preset answers. Furthermore, the inference phase execution pipeline is aligned with the training phase; the model first generates a rationale when given a question, and then, based on the question-rationale pair, predicts the final answer, further alleviating potential reasoning biases caused by spurious correlations.

Extensive experiments demonstrate that CasCoD outperforms the baselines on both IND and OOD benchmark reasoning datasets (§4.3). Besides, we validate the generalizability of CasCoD across different models, model sizes, and training data sizes (§4.4). Further analyses validate our hypothesis (§5.1) and confirm the significant impact of the two-step cascading learning process (§5.2) and the robustness of CasCoD (§5.3). The experiments on reasoning faithfulness (§5.4) and case studies (§5.5) indicate that models distilled by CasCoD can reason more consistently and demonstrate better generalization than baselines, effectively addressing interference from question-answer spurious correlations. Our contributions are as follows:

- We find that standard CoT distillation methods exhibit limited generalizability on OOD tasks, almost performing worse than methods fine-tuned directly with question-answer pairs.
- We decompose the traditional single-step learning process into two cascading learning steps to alleviate the impact of spurious correlations between questions and answers.
- Extensive experiments confirm the effectiveness of our method across both IND and OOD datasets, showing that CasCoD can generate more generalizable CoTs.

2 Related Works

CoT Capability of Language Models. LLMs have demonstrated a wide array of capabilities in numerous natural language processing tasks, underscored by various studies (Chowdhery et al., 2023; Wei et al., 2022a). One notable manifestation of this is the CoT prompting technique (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 show that SLMs' CoT performance can be further enhanced by decomposing the standard CoT distillation process into two cascaded learning steps.

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 LLMs like ChatGPT (OpenAI, 2023). These efforts typically focus on

[2] q: the question, r: the rationale, a: the answers.

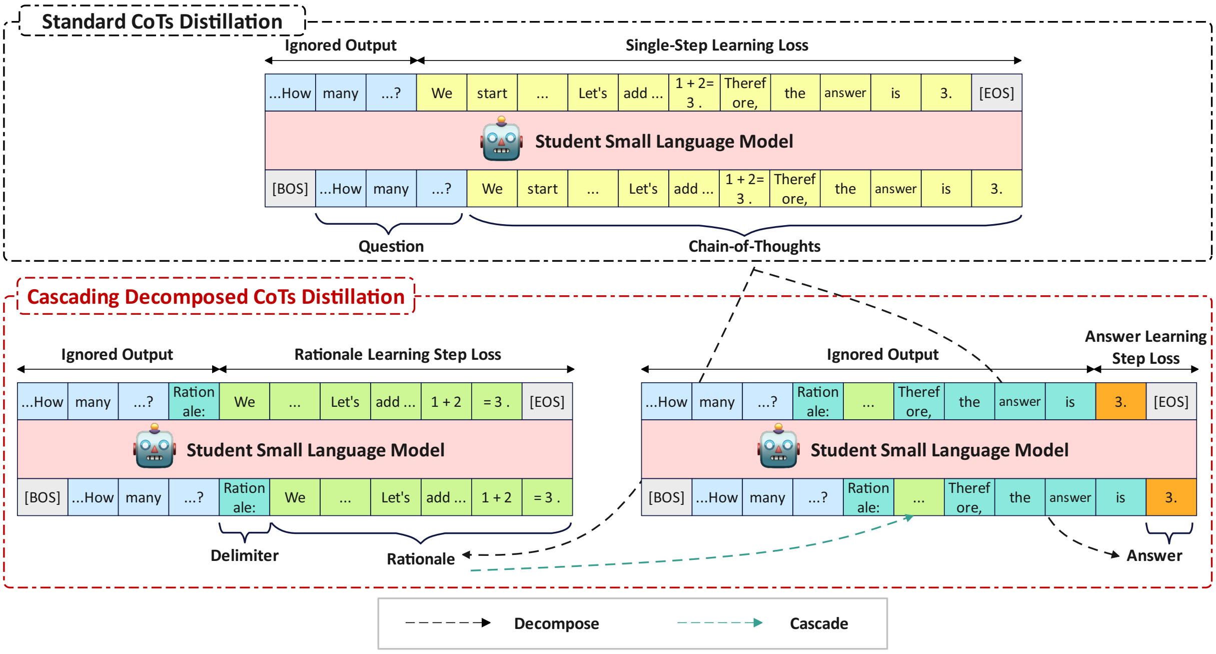


Figure 2: Overview of our proposed method **Cascading Decomposed CoTs Distillation (CasCoD)**. Different from the standard CoTs distillation, we decompose the single CoT learning step into two comprehensive learning steps including the rationale learning step and the answer learning step, and then learn them in a cascaded way.

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., 2023b; Li et al., 2024). However, our work is aimed at distilling the CoT reasoning capabilities from LLMs, in line with Magister et al. (2023); Ho et al. (2023), who propose standard CoTs distillation by directly fine-tuning SLMs on teacher LLMs' CoTs. Fu et al. (2023) expands on this by using various reasoning data formats for specializing domain-specific SLMs. Wang et al. (2023c) distill SLMs via learning from self-reflection and feedback in an interactive, multi-round paradigm with teacher LLMs. Hsieh et al. (2023) propose to learn the rationale and answers as separate goals for optimizing. Li et al. (2022) propose learning the entire CoTs and the single answers to enhance the reasoning of students. Based on these, 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. And Chen et al. (2024) maximize the mutual information between objectives. Ranaldi and Freitas (2024) use in-family and out-family teachers to generate more CoTs for fine-tuning students. However, these methods are affected by the spurious correlations due to their isolated optimization objectives. In contrast, we reorganize the training objectives to effectively mitigate this issue.

3 Methodology

In this section, we introduce our new distillation method that decomposes the single-step leanring process of standard CoTs distillation into two cascaded learning steps, as illustrated in Figure 2. Formally, the standard CoTs distillation objective $q \rightarrow CoT$ is split into two learning processes, rationale step learning with the objective $q \rightarrow r$ and answer step learning with the objective $q, r \rightarrow a$. Below we first describe how to extract CoTs from teacher LLMs in §3.1. Then we describe the standard CoTs distillation method and discuss its limitations in §3.2. Finally, we provide a detailed presentation of our method in §3.3.

3.1 Extract CoTs From Teacher LLMs

The initial phase of the distillation is to extract CoTs from teacher LLMs for each question-answer pair $\{q, a\}$ in a raw dataset. This involves employing the CoT prompting technique (Wei et al., 2022b), 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. Note that CoTs produced by LLMs may not always be correct. To maintain CoTs quality, following the previous work (Magister et al., 2023; Hsieh et al., 2023), we retain only those that match the ground truth answer in the dataset, building a CoT dataset $\mathcal{D} = \{q, CoT\}$ for training the student model. Additionally, to facilitate the introduction of CasCoD, we explicitly split the extracted CoTs into two parts based on predefined rules in CoT prompting, formalizing this as $CoT = r \oplus a$. For instance, we use the phrase "Therefore, the answer is" to divide the CoT, categorizing the text before this delimiter as the rationale r and the text after it as the answer a.

3.2 Preliminaries for CoTs Distillation

Previous standard CoTs distillation (Magister et al., 2023; Ho et al., 2023), referred to as single-step learning, is to teach SLMs to generate the CoT in one time as follows:

$$\mathcal{L}_{\text{Std-CoT}} = \mathbb{E}_{q,CoT\sim\mathcal{D}}\left[\ell\left(q, CoT\right)\right] \tag{1}$$

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

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

However, this method requires the model to simultaneously learn both rationales and answers in a single step, readily leading to the capture of question-answer spurious correlations in widespread implicit reasoning datasets (Blodgett et al., 2020). These correlations degrade the quality of CoT generation during inference, resulting in weak reasoning generalization. In other words, this implicit learning of correlations might lead student models to preset answers after reading the questions, potentially causing a state reduction in the token generation space when producing CoTs.

3.3 Cascading Decomposed CoTs Distillation

Different from the training strategy in standard CoTs distillation, our method decomposes its single-step learning process into two cascaded learning steps, one for the rationale learning step and the other for the answer learning step.

For the rationale learning step, each question is combined with a rationale learning delimiter "Rationale:" as the input q, with the rationale r produced by the teacher serving as the label for distilling the rationale. With the answer objective removed, this training strategy allows models to engage in learning rationales without the interference of spurious correlations. The loss function of rationale step learning is as follows:

$$\mathcal{L}_{\text{rationale}} = \mathbb{E}_{q,r,a\sim\mathcal{D}}\left[\ell\left(q, r\right)\right] \tag{3}$$

For the answer learning step, we concatenate both the input and output of the rationale learning step with an answer learning delimiter "Therefore, the answer is" as the input, and the answer a serves as the label for distilling the answer. This strategy helps students learn to reason consistently from the question-rationale pair rather than merely presetting spurious answers based solely on the question. The loss function of answer learning step is thus:

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

Due to the inherent tight connection between rationale learning and answer learning, for each instance in the dataset, we optimize both learning objectives simultaneously for the CoTs distillation:

$$\mathcal{L}_{\text{CasCoD}} = (1-\alpha)\mathcal{L}_{\text{rationale}} + \alpha\mathcal{L}_{\text{answer}} \tag{5}$$

where α is a hyperparameter used to weight the loss in the two learning steps.

During inference, student models follow the same pipeline as in training: first, generate a rationale based on the question, and then predict the final answer using the question-rationale pair. The cascading training objectives reduce the probability of student models capturing spurious correlations between questions and answers in the training phase, thereby alleviating potential reasoning biases caused by spurious correlations in the inference stage, thus enhancing CoTs generalizability.

4 Experiments

In this section, we conduct extensive experiments and analysis to evaluate the effectiveness of our method across both IND and OOD datasets.

4.1 Datasets

In-domain Dataset: 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. We randomly divide the BBH dataset into a training set (BBH-train) for distillation and a test set (BBH-test) for IND evaluation, in a 4:1 ratio.

Out-of-domain Datasets: (1) BIG-Bench Sub (BB-sub). BB is a popular benchmark consisting of 203 tasks covering a wide range of topics, including mathematics, common-sense reasoning, and various other domains. For ease of evaluation, we filter the subtasks within BB based on subtask keywords, focusing on tasks related to "multiple-choice" and "reasoning", 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 BB-sub. **(2) 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. **(3) 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 all above datasets can be found in Appendix B.1.

4.2 Models & Baselines & Setup

Models. We employ the popular open-source language model LLaMA2-7B (Touvron et al., 2023) as the student SLM in the main experiment and also explore different student models in §4.4. Considering the pricing and capabilities, we utilize OpenAI's powerful black-box LLM, `gpt-3.5-turbo-0613`, as teacher LLMs to extract CoTs 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; Ho et al., 2023), which is the standard CoTs distillation method that directly fine-tune student models on the CoTs data for supporting our core contributions by contrasting with our method. (3) **Step-by-step** (Hsieh et al., 2023) is a multi-task CoTs distillation method that distills rationales and answers separately. (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 entire CoTs learning. These methods, which also modify training strategies, are compared to illustrate the superiority of our approach in decomposing CoTs and cascading learning. (5) **SCOTT** (Wang et al., 2023a) that enhances the reasoning consistency of the student model by introducing additional counterfactual data. This method is chosen to assess whether additional negative examples can enhance reasoning performance and to compare student reasoning faithfulness in subsection 5.4.

Setup. We employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning of the student SLMs. In §5.3, our empirical results indicate that the optimal weight is set α at 0.3. However, to mitigate the effects of unbalanced weighting, we include an additional method setup for comparison against the baselines in Table 1, labeled CasCoD ($\alpha = 0.5$). All experiments are conducted using a mixed-precision training strategy on 4 $\times$ A100 GPUs. For the inference stage, vLLM[3](Kwon et al., 2023) is utilized to accelerate inference, employing a greedy decoding strategy to generate text on one single A100 GPU. More details on training and hyperparameters can be found in Appendix B.2.

4.3 Main Results

Table 1 presents the automatic evaluation results of our proposed CasCoD and baselines.

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 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 1, 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 as introduced in Figure 1 (b), which students can easily learn. The distillation baselines that require students to consider predicting answers while generating the rationale, inadvertently make the simpler task of answer prediction interfere with

[3] 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)	✓	✓	<u>56.8</u>	30.3	22.0	49.4	38.2	39.3
Step-by-step (Hsieh et al., 2023)	✓	✓	42.4	27.7	**28.8**	68.5	48.6	43.2
CasCoD (ours, $\alpha = 0.5$)	✓	✓	52.5	<u>36.4</u>	28.1	**71.8**	**54.7**	<u>48.7</u>
CasCoD* (ours, $\alpha = 0.3$)	✓	✓	**59.4**	**37.0**	<u>28.3</u>	<u>70.6</u>	<u>52.7</u>	**49.6**

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**, and the second-best performance is indicated by an <u>underline</u>.

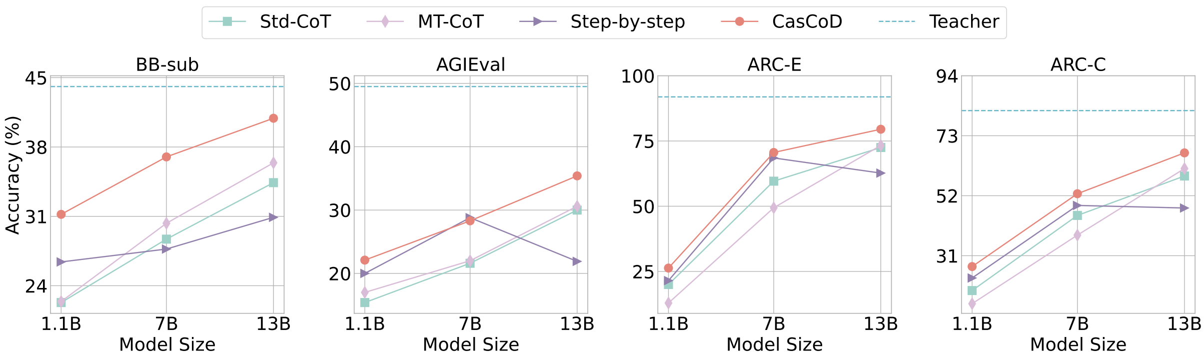


Figure 3: Ablation study on model size for four OOD datasets. The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting. The results in IND dataset can be found in Appendix A.2.1.

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 cascade two step learning strategy, which restructures training objectives to mitigate the impact of spurious correlations, 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 significantly outperforms baselines on both IND and OOD datasets in most cases, regardless of whether the loss is weighted. 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 on Model & Data Sizes

CasCoD is universally applicable to models of varying sizes. We perform model distillation on TinyLLaMA-1.1B[4] (Zhang et al., 2024), LLaMA2-7B, and LLaMA2-13B, respectively and compare

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

Models & Methods	BBH-test	BB-sub	AGIEval	ARC-E	ARC-C	AVG
In-domain?	✓	×	×	×	×	
CodeLLaMA-7B + Std-CoT	**56.2**	29.7	19.2	42.0	32.2	35.9
CodeLLaMA-7B + Step-by-step	40.7	29.0	23.9	41.5	**32.8**	33.6
CodeLLaMA-7B + CasCoD	54.8	**35.4**	**25.8**	**42.9**	31.7	**38.1**
LLaMA3-8B + Std-CoT	**66.9**	33.9	32.7	69.8	60.2	52.7
LLaMA3-8B + Step-by-step	44.2	35.5	38.8	83.7	70.7	54.5
LLaMA3-8B + CasCoD	65.2	**42.9**	**40.1**	**87.2**	**74.0**	**61.9**
Mistral-7B-v0.2 + Std-CoT	**72.2**	37.6	32.0	68.8	57.9	53.7
Mistral-7B-v0.2 + Step-by-step	56.4	38.9	20.1	76.4	62.3	50.8
Mistral-7B-v0.2 + CasCoD	71.7	**42.5**	**40.1**	**83.9**	**74.2**	**62.5**

Table 2: Accuracy (%) on IND and OOD datasets with different student models distilled by different methods.

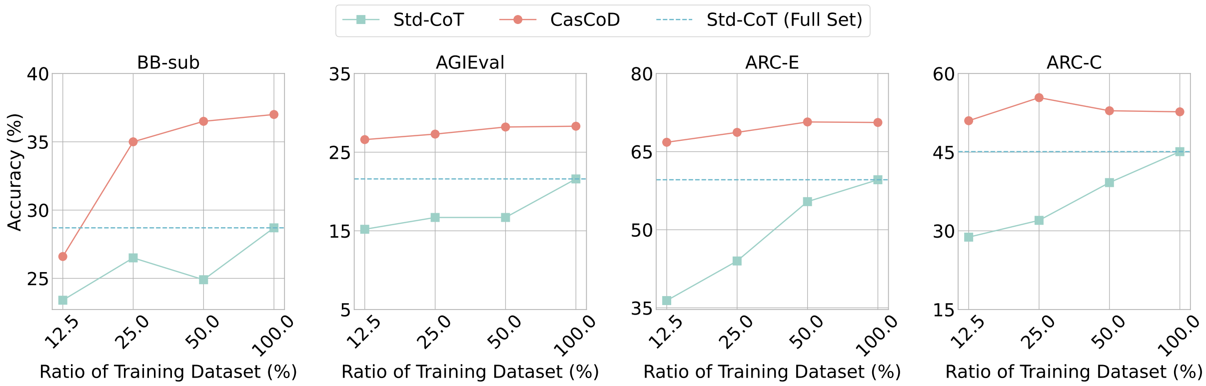


Figure 4: Ablation study on training data size for four OOD datasets. The dotted line indicates the performance of fine-tuning the student models by standard CoTs distillation using the full set (100% of) BBH-train dataset. The results in IND dataset can be found in Appendix A.2.2.

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 is universally applicable to models of different architectures. We perform model distillation on CodeLLaMA-7B (Touvron et al., 2023), LLaMA3-8B (AI@Meta, 2024) and Mistral-7B-v0.2 (Jiang et al., 2023a), respectively, and compare with Std-CoT and Step-by-step. From the Table 2, we can see that regardless of whether it's CodeLLaMA, LLaMA3, or Mistral, CasCoD significantly outperforms the baselines on OOD tasks, demonstrating its high effectiveness and scalability. Particularly, on the powerful base model Mistral, the superiority of our method is further amplified.

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.

5 Analysis

5.1 Hypothesis Validation

In this subsection, we aim to validate our hypothesis that student learning of spurious correlations

Method & Metric	BB-sub	AGIEval	ARC-E	ARC-C
Std-CoT (**A**) ↓	79.7	82.2	63.7	50.8
Std-CoT (**B**) ↑	47.5	30.6	58.1	69.7
Std-CoT (**C**) ↓	32.2	51.6	5.6	-18.9
CasCoD (**A**) ↓	73.3	77.9	61.8	50.9
CasCoD (**B**) ↑	57.4	36.7	71.4	83.9
CasCoD (**C**) ↓	**15.9**	**41.2**	**-9.6**	**-33.0**

Table 3: Hypothesis validation results (%) on four OOD tasks. The hypothesis is better supported when '**A**' is lower, '**B**' is higher, and '**C**' is lower.

affects the quality of their generated rationales. Due to the nature of causal language models—autoregression, we incorporate "The answer is" into input prompts to directly prompt models to provide the answer immediately (which we treat as the "preset answer" referred to §1) after reading the question, rather than generating rationales before providing an answer. We define the following metrics to assist the validation: (1) **A**: The proportion of incorrect CoT reasoning when the preset answer is wrong. A higher ratio indicates a greater negative impact of the preset answer on CoT reasoning; (2) **B**: The proportion of correct CoT reasoning when the preset answer is correct. A higher ratio indicates a greater positive impact of the preset answer; (3) $\mathbf{C} = \mathbf{A} - \mathbf{B}$ (combine $\mathbf{A}$ and $\mathbf{B}$). We compare CasCoD with Std-CoT using these metrics on four OOD tasks. The results are shown in Table 3. We observe that Std-CoT significantly outperforms CasCoD in metrics '**A**' and '**C**', while significantly underperforms CasCoD in metric '**B**' on all four OOD tasks. This indicates that our method can selectively utilize spurious correlations to some extent, suppressing the negative effects of incorrect preset answers on reasoning and reinforcing the positive effects of correct preset answers on reasoning, thereby enhancing performance on OOD tasks, which experimentally validate our hypothesis.

5.2 Two-Step vs. Single-Step Implementation

In this subsection, we explore whether CasCoD's two-step training objectives can be achieved in a single-step computation. Upon analysis of the two cascaded learning steps, we find that under teacher-forcing (Goodfellow et al., 2016), CasCoD closely mirrors Std-CoT, with key distinctions including adjustable token-level weights and the omission of delimiters in loss calculations. Each sample in CasCoD's original framework undergoes two forward calculation, raising the question of whether a similar outcome is possible with only one. To investigate this, we introduce a variant, CasCoD-single, which is designed to fulfill the two-step training objectives through a single forward computation. Figure 5 indicates that the two-step CasCoD consistently surpasses the single-step variant across all datasets. This underscores that a single forward calculation does not suffice to meet CasCoD's training objectives, emphasizing the critical importance of the cascading two-step learning process.

5.3 Impact of Weights

In this subsection, we explore how variations in weights affect the performance of models 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 range approximately from 0.01 to 0.3, indicating that focusing on the rationale help improve the generalizability of CoTs.

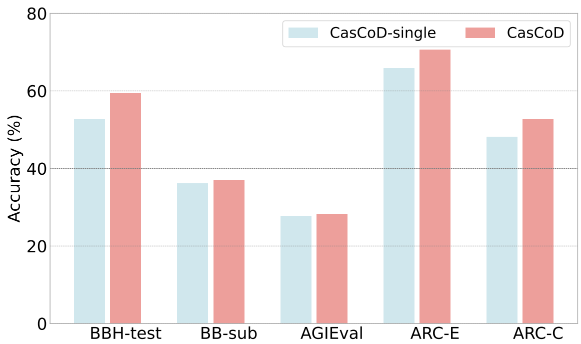


Figure 5: Comparison between two-step and single-step training implementations of CasCoD.

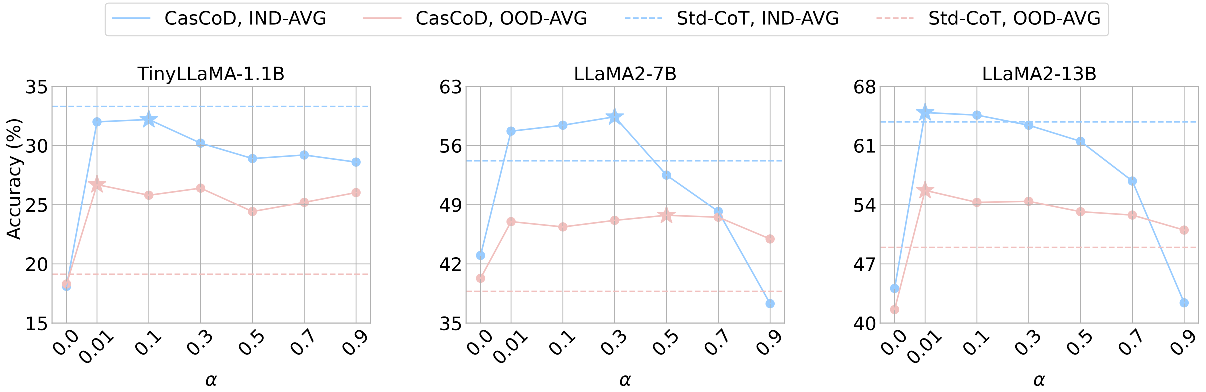


Figure 6: Ablation study on task weights α. The results are reported by IND-AVG and OOD-AVG that respectively denote averge accuracy on IND and OOD datasets. The best performance among weights are marked with "☆".

5.4 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:

$$LAS = Acc\left(q, r' \rightarrow a'\right) - Acc\left(q \rightarrow a'\right) \quad (6)$$

where we employ ChatGPT and GPT4 as the simulator, respectively. The results are shown in Table 4. CasCoD is observed to generate rationales that are more consistent with answers than baselines. This suggests that despite CasCoD's multi-step learning process, the introduction of cascading learning ensures that students can faithfully reason.

Method	ChatGPT	GPT4	AVG
Teacher	41.1	36.3	38.7
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 4: Faithfulness (LAS, %) of the compared methods with different LLM evaluators on the IND dataset. The prompt templates can be found in Appendix C.2

.

5.5 Case Study

In this subsection, we provide a systematic case study to illustrate our hypothesis and the improvement in CoT generalizability. Here we show 4 cases in Table 17, 18, 19 and 20 to compare the CoT generated by CasCoD with the teacher LLM and the standard CoTs distillation method (Std-CoT). We utilize ✓ and ✗ to denote whether the CoT is correct or incorrect, respectively.

Table 17 and 18 show that while Std-CoT correctly predicts the final answer on in-domain tasks, it generates incorrect intermediate reasoning steps, indicating that Std-CoT causes student models to capture spurious correlations between questions and answers, without learning to reason with diversity. This is evidenced by the fact that its generated rationales are almost identical to the teacher's CoTs. In contrast, CasCoD produces more distinct CoTs, differing from the teacher's reasoning format, which indicates an enhancement in the student's reasoning generalizability. Tables 19 and 20 illustrate that on out-of-domain tasks, Std-CoT fails to reason correctly, including logical and factual mistakes, whereas CasCoD can clearly understand the question and provides concise, logically rich reasoning. Besides, we also provide an analysis of failure cases in Appendix A.4.

6 Conclusion

We propose a simple yet effective CoTs distillation method CasCoD to address the issue of question-answer spurious correlations that previous CoTs distillation methods suffer from. Specifically, we decompose the traditional single-step learning process into two cascaded learning steps and restructure their training objectives. Extensive experiments show that CasCoD significantly outperforms the baselines across both IND and OOD datasets. Further analysis reveals that CasCoD is robust to model size, training data size, different models, and weights and can lead to faithful student models.

Limitations

In our study, we explore distilling CoTs into two cascading steps, which is an initial step toward understanding finer decompositions. Research (Schaeffer et al., 2023a) suggests that the emergent abilities of LLMs result from managing multiple subtasks simultaneously, hinting at the potential for more intricate cascading steps in CoTs. Our current work does not yet define the precise rules for such more steps decomposition, nor the optimal timing and methods for focusing learning on specific steps. We hope our work can inspire the community and leave these aspects for future exploration, intended to refine and extend the CoT reasoning capabilities of SLMs as suggested by our findings.

Ethics Statement

Our 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

AI@Meta. 2024. Llama 3 model card.

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.

Xin Chen, Hanxian Huang, Yanjun Gao, Yi Wang, Jishen Zhao, and Ke Ding. 2024. Learning to maximize mutual information for chain-of-thought distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*.

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.

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.

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.

Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2022. Thinking fast and slow in large language models. *arXiv preprint arXiv:2212.05206*.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *CoRR*, abs/2310.06825.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023b. 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*.

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.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. *CoRR*, abs/2402.10110.

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.

Leonardo Ranaldi and André Freitas. 2024. Aligning large and small language models via chain-of-thought reasoning. In *EACL (1)*, pages 1812–1827. Association for Computational Linguistics.

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*.

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.

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.

Yuhang Zhou and Wei Ai. 2024. Teaching-assistant-in-the-loop: Improving knowledge distillation from imperfect teacher models in low-budget scenarios.

A Additional Experimental Results

A.1 Detailed Results on Mathematical Reasoning Tasks

Given the community's focus on mathematical reasoning, we present detailed experimental results for math-related subtasks from our evaluation datasets in Table 5. It should be noted that the subtasks in the datasets we utilized involve mathematical reasoning tasks. We list as follows:

1. BB-sub: includes subtasks like 'elementary_math_qa' and 'identify_math_theorems', as detailed in Table 11.

2. BBH: includes subtasks such as 'DateUnderstanding' and 'Multi-StepArithmetic', as detailed in Table 10.

3. AGIEval: includes subtasks like 'AQuA-RAT' and 'SAT-Math', as detailed in Table 8.

While ARC does not specifically feature tasks for mathematical computation, we identify a number of scientific questions within this dataset that involve mathematical calculations. From the Table 5, We can observe that CasCoD has achieved performance improvements on mathematical reasoning tasks as well.

A.2 Ablation Study on In-domain Dataset

A.2.1 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.

A.2.2 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 CoTs distillation across various sizes of training data. This demonstrates the efficiency of our proposed method.

Models & Tasks	elementary_math_qa	identify_math_theorems	DateUnderstanding	Multi-StepArithmetic	AVG
Std-CoT	11.0	20.7	82.0	8.0	30.4
SCOTT	12.0	34.0	58.0	4.0	27.0
MT-CoT	10.0	9.4	74.0	6.0	24.9
Step-by-step	17.0	32.1	68.0	0.0	29.3
CasCoD	**22.0**	**36.2**	**86.0**	**8.0**	**38.1**

Table 5: Accuracy (%) on mathematical reasoning tasks with different distillation methods.

Rationale Delimiter	Answer Delimiter	BBH-test	BB-sub	AGIEval	ARC-E	ARC-C	AVG
`abc:`	`xyz:`	55.3	33.8	26.4	69.5	52.0	47.4
`cat:`	`dog:`	50.4	35.3	25.6	66.3	50.8	45.7
`Reason:`	`Draw a conclusion:`	56.3	35.7	25.4	70.4	51.7	47.9
`Rationale:`	`Answer:`	59.4	37.0	28.3	70.6	52.7	49.6

Table 6: Performance comparison of different delimiters across several benchmarks.

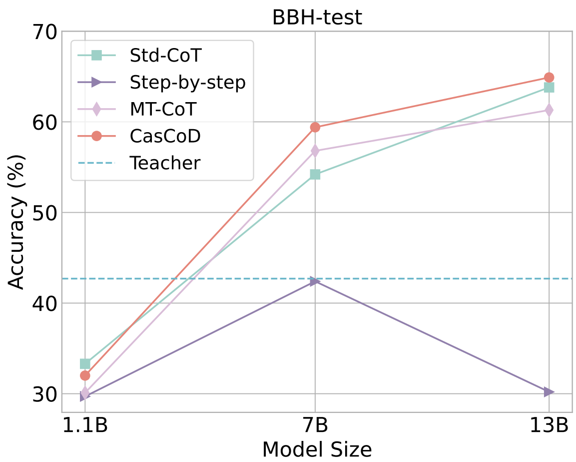


Figure 7: Ablation study on model size in the IND (BBH-test). The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting.

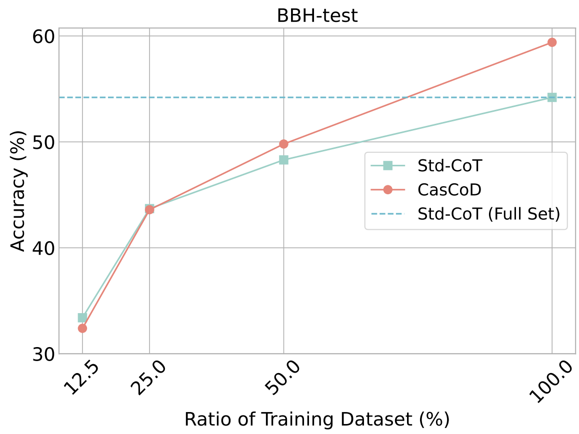


Figure 8: Ablation study on training data size in the IND (BBH-test). The dotted line indicates the performance of fine-tuning the student models by standard CoT distillation using the full set (100% of) BBH-train dataset.

A.3 Impact of Delimiters

In NLP models, delimiters serve as cues to guide the model's reasoning and answer generation processes. Initially, we designed our system to use specific, semantically clear delimiters such as `Rationale:` to signal the reasoning phase and `Therefore, the answer is:` to indicate the final conclusion. This design aimed to maintain a natural flow and avoid semantic confusion. However, it remained important to evaluate whether the choice of delimiters might unintentionally influence the model's performance or introduce bias. Therefore, we conduct experiments using a range of different delimiters, categorized into three types:

- **Random, non-semantic sequences of letters:** We use arbitrary sequences such as `"abc:"` and `"xyz:"` to test if non-semantic strings impact model performance.
- **Random words:** Common, unrelated words like `"cat:"` for rationale and `"dog:"` for answer are used to see if semantically irrelevant words influence performance.
- **Phrases with similar semantics:** For this category, we select phrases like `"Reason:"` for the reasoning step and `"Draw a conclusion:"` for the answer, as these phrases align more closely with the logical structure of the tasks.

Table 6 presents the performance of each delimiter type. We report the average performance across all tasks. From the results, we observe that the semantic delimiters `"Rationale:"` and `"Answer:"` achieved the best average performance (49.6%), showing that clear and intuitive cues improve model outcomes. The non-semantic delimiters "abc:" and "xyz:" resulted in moderate performance, indicating that the model can function reasonably well with arbitrary signals, though less effectively. Interestingly, random words like "xyz:" and "dog:" yielded the lowest performance (45.7%), suggesting the model might unintentionally ascribe meaning to these words, affecting reasoning. Phrases such as "Reason:" and "Draw a conclusion:", which align with the model's learning structure, performed nearly as well as the default semantic delimiters. While delimiters don't drastically affect performance, semantically meaningful ones yield better results. Non-semantic or random words can still function but tend to reduce reasoning clarity, especially in more complex tasks.

A.4 Analysis of Failure Cases

We conducted a detailed error analysis, particularly focusing on the performance of our model CasCoD on OOD tasks. The evaluation involved precision, recall, and F1 scores to better understand CasCoD's failure cases and its comparison to baselines. Below we present both quantitative and qualitative analysis of these failure cases.

A.4.1 Quantitative Analysis

We compared CasCoD with baselines on three challenging OOD subtasks: *identify_math_theorems* (4-class), *social_iqa* (3-class), and *LSAT-LR* (5-class) from the BB-sub and AGIEval datasets. The results are summarized in Table 7. These metrics provide a more nuanced view than accuracy alone and help identify where CasCoD struggles the most. The quantitative results in Table 7 suggest that CasCoD struggles to achieve competitive precision and recall compared to other models across these tasks. For instance, in the *identify_math_theorems* task, CasCoD demonstrates moderate accuracy (36.2%) but falls behind in both precision (26.4%) and recall (24.2%), indicating difficulty in making precise distinctions between classes. In the *social_iqa* task, CasCoD's performance is significantly weaker, with an F1 score of only 7.6%. For the *LSAT-LR* task, CasCoD also underperforms across all metrics, indicating that improvements are needed in reasoning about complex arguments.

A.4.2 Qualitative Analysis

We analyzed the false examples from three underperforming tasks: `identify_math_theorems`, `social_iqa`, and `LSAT-LR`.We show the details of each task in Table 19, Table 22 and Table 23, including task description, an example question and CasCoD's CoTs, along with our analysis.

Task: `identify_math_theorems` This task involves verifying mathematical theorems and correcting them if necessary. CasCoD incorrectly selected option (D) instead of the correct answer (B), suggesting that the model struggles with distinguishing nuanced differences in mathematical notation and formulations. This confusion is particularly apparent in complex mathematical expressions, where slight changes in notation can lead to drastically different conclusions.

Task: `social_iqa` This task requires the model to understand the outcomes of social interactions. In this example, CasCoD chose (C) instead of the correct answer (A), indicating its difficulty in discerning the source of an emotional response and its direct manifestations. This suggests that the model may need more training data that focuses on distinguishing subtle emotional cues in social contexts.

Task: `LSAT-LR` This task involves analyzing assumptions in logical arguments. In this case, CasCoD selects option (C), but the correct answer is (D). The error suggests that the model has difficulty identifying the correct logical principles underlying complex arguments, indicating that its logical reasoning capabilities require further refinement and more targeted training on complex argument structures.

B Details of Experiment

B.1 Dataset Statistics

Table 8, Table 9, Table 10 and Table 11 show the data statistics of AGIEval, ARC, BIG-Bench Hard (BBH) and BIG-Bench Sub (BB-sub)[5], respectively.

B.2 Hyperparameters Settings

In our study, we ensure consistency in the hyperparameter settings across all baselines, including our proposed CasCoD approach, to maintain the fairness of our comparative analysis. Here, we detail the hyperparameter configurations employed in our experiments.

[5]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.

Task	Method	Precision	Recall	F1 Score	Accuracy
identify_math_theorems	Std-CoT	37.9	39.6	31.7	40.4
	SCOTT	25.0	25.0	20.6	31.9
	MT-CoT	23.6	16.1	17.9	25.5
	Step-by-step	36.1	33.7	31.9	14.9
	CasCoD	26.4	24.2	25.1	36.2
social_iqa	Std-CoT	45.8	46.2	45.3	35.0
	SCOTT	26.0	23.3	22.9	20.0
	MT-CoT	48.5	52.4	47.7	30.0
	Step-by-step	42.4	43.3	42.5	41.0
	CasCoD	8.0	7.4	7.6	19.0
LSAT-LR	Std-CoT	27.0	26.2	26.1	20.0
	SCOTT	26.6	26.9	26.3	15.3
	MT-CoT	33.6	34.5	33.9	24.7
	Step-by-step	39.9	34.9	32.6	30.3
	CasCoD	23.7	23.2	23.2	22.2

Table 7: Performance comparison of CasCoD and baselines on three OOD subtasks using Precision, Recall, F1 Score, and Accuracy metrics.

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 8: Statistics of AGIEval dataset.

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

Table 9: 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 10: Statistics of BIG-Bench Hard dataset.

Training Steps and Batch Size. The number of training steps is determined based on the size of the training dataset, the batch size, and the number of gradient accumulation steps required. We maintain a consistent batch size across all baselines to eliminate any performance discrepancies that could arise from varying batch sizes.

Learning Rate. Our exploratory experiments initially focus on the standard CoTs distillation method using the LLaMA-2 model, revealing that while the batch size had minimal impact on performance, the learning rate was a critical factor. We test learning rates of 1e-4, 2e-4, and 3e-4 and observe optimal performance at 2e-4 across Std-CoT and other distillation baselines as well as our CasCoD. Therefore, we set the learning rate to 2e-4 for all methods involved in our study.

Epochs and Evaluation Strategy. Throughout our training process, we monitor the training loss curve and note that it generally plateaued by the 15th epoch, suggesting that the models have achieved convergence. Accordingly, we set the number of epochs to 15 for 7B models. The process of determining the number of epochs for other model sizes follows a similar pattern. To mitigate the potential risk of overfitting and to ensure that our evaluation reflects the most effective model configuration, we systematically select the check-

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 11: 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.

points from the epoch that demonstrate the best performance on the IND task. These checkpoints are then used to evaluate performance on OOD tasks.

Finally, the detailed hyperparameters in training and inference can be found in Table 12 and Table 13, respectively.

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 12: 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 13: Generation configs of students and teachers.

C Prompts

C.1 Prompts of Generating CoTs for ChatGPT

We use the prompt template shown in Table 14 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 15 and Table 16 to call the ChatGPT and GPT4 API to predict the answers given a question or with an additional rationale, respectively.

```
{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 No.1}.

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

Q: {Task Example Question No.3}
A: Let's think step by step. {Human-Curated-CoTs No.3}.

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

Table 14: Prompt template of gpt-3.5-turbo for generating the CoTs data with 3 shots.

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 15: 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 16: Prompt template of simulators for predicting the answers when given the question and rationale.

Question:	Evaluate the result of a random Boolean expression: not False and False or True or False.
Teacher	Rmember that (i) expressions inside brackets are always evaluated first and that (ii) the order of operations from highest priority to lowest priority is "not", "and", "or", respectively. We first simplify this expression "Z" as follows: "Z = not False and False or True or False = A and B or C or D" where "A = not False", "B = False", "C = True", and "D = False". Let's evaluate A: A = not False = not (False) = True. Plugging in A, B, C, and D, we get: Z = A and B or C or D = True and False or True or False = False or True or False = True. Therefore, the answer is True. ✓
Std-CoT	Rmember that (i) expressions inside brackets are always evaluated first and that (ii) the order of operations from highest priority to lowest priority is "not", "and", "or", respectively. We first simplify this expression "Z" as follows: "Z = not False and False or True or False = A and B or C or D" where "A = not False", "B = False", "C = True", and "D = False". Let's evaluate A: A = not False = not (False) = True. Plugging in A, B, C, and D, we get: Z = A and B or C or D = True and False or True or False = False or False or False = False. Therefore, the answer is True. ✗
CasCoD	Removing the brackets first: (not False and False) or (True or False) or False. Let's evaluate these expressions one by one: - (not False and False): The boolean expression "not False and False" evaluates to True. - (True or False): The boolean expression "True or False" evaluates to True. - False: The boolean expression False evaluates to False. When we evaluate these expressions, we get True. Therefore, the answer is True. ✓

Table 17: A boolean-expression case from BIG-Bench Hard.

Question:	Answer extremely simple questions about the colors of objects on a surface. Q:On the table, you see several objects arranged in a row: a burgundy paperclip, a purple cup, a mauve pencil, a yellow keychain, and a silver teddy bear. What is the color of the object furthest from the teddy bear? Options: (A) red (B) orange (C) yellow (D) green (E) blue (F) brown (G) magenta (H) fuchsia (I) mauve (J) teal (K) turquoise (L) burgundy (M) silver (N) gold (O) black (P) grey (Q) purple (R) pink
Teacher	According to this question, the objects are arranged in a row, from left to right, as follows: (1) a burgundy paperclip, (2) a purple cup, (3) a mauve pencil, (4) a yellow keychain, (5) a silver teddy bear. The teddy bear is the last item on the list, namely (5). The object furthest from the teddy bear corresponds to (1), which is a burgundy paperclip. The color of the paperclip is burgundy. Therefore, the answer is (L).. ✓
Std-CoT	According to this question, the objects are arranged in a row, from left to right, as follows: (1) a burgundy paperclip, (2) a purple cup, (3) a mauve pencil, (4) a yellow keychain, (5) a silver teddy bear. The object furthest from the teddy bear is the mauve pencil, which is the third item on the list. The color of the mauve pencil is mauve. Therefore, the answer is (L). ✗
CasCoD	Rationale: According to this question, the objects are arranged in a row, from left to right, as follows: (1) a burgundy paperclip, (2) a purple cup, (3) a mauve pencil, (4) a yellow keychain, (5) a silver teddy bear. The object furthest from the teddy bear corresponds to (1), which is a burgundy paperclip. The color of the paperclip is burgundy. Therefore, the answer is (L). ✓

Table 18: A reasoning about colored objects case from BIG-Bench Hard.

Question:	$h = 3a + 28.6$ A pediatrician uses the model above to estimate the height h of a boy, in inches, in terms of the boy's age a, in years, between the ages of 2 and 5. Based on the model, what is the estimated increase, in inches, of a boy's height each year? Options: (A)3 (B)5.7 (C)9.5 (D)14.3
Teacher	Step 1: We are given the model $h = 3a + 28.6$ which represents the estimated height h of a boy in terms of his age a. Step 2: We need to find the estimated increase in height each year. This means we need to find the change in height for a change in age of 1 year. Step 3: To find the change in height, we can subtract the height at age $a + 1$ from the height at age a. So, the change in height is $(3(a + 1) + 28.6) - (3a + 28.6)$. Step 4: Simplifying the expression, we get $3a + 3 + 28.6 - 3a - 28.6$. Step 5: The terms $3a$ and $-3a$ cancel out, leaving us with $3 + 3$. Step 6: The change in height is 6 inches. Therefore, the answer is (A) 3. ✗
Std-CoT	This question focuses on the increase in height each year. According to the model, the height of a boy between the ages of 2 and 5 is approximately 36 inches. Therefore, the estimated increase in height each year is 3 inches. Therefore, the answer is (A). ✗
CasCoD	Rationale: This question asks about the estimated increase, in inches, of a boy's height each year. According to the model, the height of a boy increases by 3 inches each year between the ages of 2 and 5. Therefore, the answer is (A) 3 inches. ✓

Table 19: A sat-math case from AGIEval.

Question:	Which gas is the most abundant in Earth's atmosphere? Options: (A) oxygen (B) nitrogen (C) water vapor (D) carbon dioxide
Teacher	First, we know that Earth's atmosphere is primarily composed of gases. Second, we need to determine which gas is the most abundant. Third, we can eliminate options (A) oxygen and (C) water vapor because they are present in smaller quantities compared to the other gases. Finally, we are left with options (B) nitrogen and (D) carbon dioxide. Considering that nitrogen makes up about 78% of Earth's atmosphere, while carbon dioxide only makes up about 0.04%, it is clear that nitrogen is the most abundant gas in Earth's atmosphere. Therefore, the answer is (B) nitrogen. ✓
Std-CoT	The percentage of each gas in the atmosphere is given by the following table: Gas Percentage oxygen 20.95 nitrogen 78.09 water vapor 0.4 carbon dioxide 0.04 According to this table, oxygen is the most abundant gas in Earth's atmosphere. Therefore, the answer is (A). ✗
CasCoD	Rationale: According to the World Bank, the composition of Earth's atmosphere is approximately 78% nitrogen, 21% oxygen, 0.93% argon, and small amounts of other gases. Nitrogen is the most abundant gas in Earth's atmosphere. Therefore, the answer is (B). ✓

Table 20: A case from AI2 Reasoning Challenge.

Description	The task involves verifying mathematical theorems and correcting them if false.
Example Question	What follows is a purported mathematical theorem. Some will be true, while other will be false. If the theorem is correct, write the theorem exactly as it is given. Otherwise, write a corrected version of the theorem. Write all answeres in compilable LaTeX. **Theorem:** Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{R})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{\widehat{G}}$. Options: (A) Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{R})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{G}$. (B) Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{T})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{\widehat{G}}$. (C) Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{C})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{\widehat{G}}$. (D) Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{R})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{\widehat{G}}$. **Corrected Theorem:**
CasCoD's CoT	Here is the correct version of the theorem: Let G be a locally compact abelian group and let $\widehat{G} := Hom(G, \mathbb{R})$ be its Pontryagin dual. There is a canonical isomorphism between G and $\widehat{\widehat{G}}$. Therefore, the answer is (D).

Table 21: A failure case on identify_math_theorems task.

Description	This task asks about probable outcomes in social interactions.
Example Question	Sasha gave the team a sense of inspiration when they gave a pep talk. How would Others feel as a result? Options: (A) Full of spirit (B) Like they are ready to do their best (C) Very motivating
CasCoD's CoT	Here in this question, we are told that Sasha gave the team a sense of inspiration when they gave a pep talk. This suggests that Sasha's presence and words had a positive impact on the team and helped to motivate and inspire them. As a result, Others would likely feel like they are ready to do their best and feel motivated and inspired by Sasha's presence. Therefore, the answer is (C).

Table 22: A failure case on social_iqa task.

Description	This task asks about probable outcomes in social interactions.
Example Question	Passage: Chairperson: The board of directors of our corporation should not allow the incentives being offered by two foreign governments to entice us to expand our operations into their countries without further consideration of the issue. Although there is an opportunity to increase our profits by expanding our operations there, neither of these countries is politically stable. Question: The chairperson's reasoning most closely conforms to which one of the following principles? Options: (A)A corporation should never expand operations into countries that are politically unstable. (B)Corporations should expand operations into countries when there is a chance of increasing profits. (C)Political stability is the most important consideration in deciding whether to expand operations into a country. (D)Corporations should always be cautious about expanding operations into politically unstable countries. (E)Boards of directors should always disregard governmental incentives when considering where to expand corporate operations.
CasCoD's CoT	Here in this question, the chairperson expresses concern about the political stability of the two foreign countries and suggests that the incentives being offered by the governments should be further considered before making a decision. The principle that aligns most closely with this reasoning is (D), which states that corporations should always be cautious about expanding operations into politically unstable countries. This principle acknowledges the potential risks and challenges associated with operating in countries with political instability and recommends caution and careful consideration before making a decision. Therefore, the answer is (C).

Table 23: A failure case on LSAT-LR task