Mistake-assisted Distillation: Enhancing Student's CoT Capabilities by Identifying Key Reasoning Steps

Anonymous ACL submission

Abstract

With the scaling up of model parameters, powerful reasoning capabilities have emerged in Large Language Models (LLMs). However, resource constraints in practical applications pose challenges to the deployment of such models, which prompted a lot of attention to distilling the capabilities into smaller, compact language models. Prior distillation works simply fine-tune student models on Chain-of-Thoughts (CoTs) data generated by teacher LLMs, resulting in the student merely imitating the teacher's reasoning style without capturing the key in reasoning. In this paper, we propose a novel distillation method called **Mis**take-**A**ss**i**sted **D**istillation (MisAiD) to help students identify the key reasoning steps and learn the thinking way in reasoning. Specifically, we first retain all CoT data annotated by teacher LLMs, irrespective of correctness. Then, we design specific prompts to rectify teachers' wrong CoTs and mistake the correct CoTs, respectively, forming the dual CoTs data that have similar reasoning steps but divergent conclusions. Finally, we identify the key reasoning steps in dual CoTs and employ a fine-grained loss function to guide student learning. Extensive experiments and comprehensive analyses demonstrate the effectiveness of MisAiD on both in-domain and out-of-domain benchmark reasoning datasets.

1 Introduction

With the rapid growth in model size and pre-training data, LLMs have demonstrated impressive performance in natural language processing (NLP) (Brown et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2023; OpenAI, 2023b). However, due to the giant model architecture and massive parameters, the deployment of LLMs in resource-constrained environments becomes challenging.

To address this, researches (Xu et al., 2023; Jiang et al., 2023) have explored distilling knowledge from LLMs into smaller language models (SLMs) via instruction-tuning, as seen in LMs like Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023). Despite progress, these distilled models often struggle with complex causal reasoning. To enhance this capability, some studies (Magister et al., 2023; Ho et al., 2023; Fu et al., 2023) explore distilling the CoT reasoning ability from LLMs of over 100B parameters (Wei et al., 2022a,b) by fine-tuning on CoTs data annotated by teacher LLMs, known as standard CoTs distillation. Besides, other studies (Hsieh et al., 2023; Li et al., 2022; Liu et al., 2023) propose distilling CoTs within a multi-task learning framework by incorporating additional objectives. However, the essence of the above methods is the simplistic imitation learning paradigm (Gudibande et al., 2023), where the student model is fine-tuned solely on the teacher's correct reasoning data. This

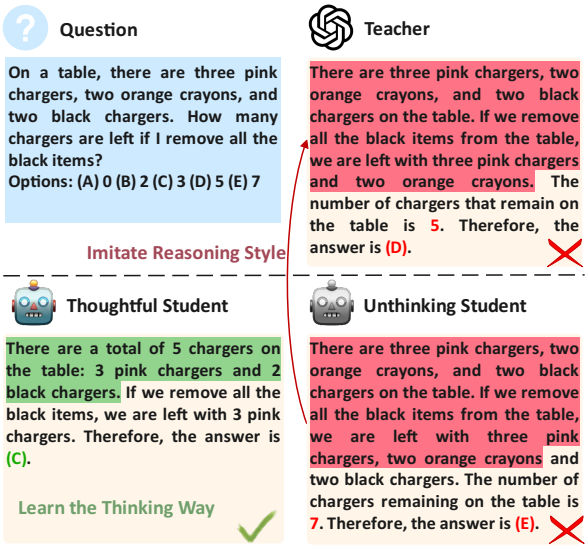

Figure 1: Examples of CoTs generated by student SLMs. Simply imitation learning leads to an unthinking student who imitates the teacher's reasoning style, where the imitated contents are highlighted in red. Our method distills a thoughtful student who can learn the thinking way from the teachers' mistakes, with these thoughtful reasoning steps are highlighted in green.

paradigm may result in the student SLM **imitating the teacher's reasoning style without learning the thinking way in reasoning**, as illustrated in Figure 1. We observe that the reasoning process of the unthinking student is almost as lengthy and complex as the teacher's, while the thoughtful student adopts a different thought, simplifying the reasoning. Therefore, we posit that students' CoT capabilities depend on their ability to understand the reason behind teachers' successes and failures.

Drawing an analogy to human learning, where analyzing mistakes often reveals key reasoning steps[1], we propose a novel **Mis**take-**A**ss**i**sted **D**istillation method (MisAiD). This approach focus on dual CoTs data, encompassing both positive and negative examples of teachers' reasoning. By examining dual CoTs, students can identify and learn from the crucial reasoning steps, thereby improving their CoTs. Specifically, we first retain all CoTs data annotated by the teacher, irrespective of correctness. Subsequently, we design two comprehensive prompts to instruct teachers to produce dual CoTs that share similar intermediate reasoning steps but lead to divergent conclusions. Finally, we utilize the minimum edit distance algorithm to identify key reasoning steps in dual CoTs, as shown in Figure 3, and then apply a fine-grained loss function for guided learning.

Extensive experiments demonstrate that the student model distilled by MisAiD exhibits higher performance and generalization than the baselines on both in-domain (IND) and out-of-domain (OOD) benchmark reasoning datasets. Further analyses indicate that MisAiD can generate higher-quality CoTs by auto evaluation and case studies. Our contributions can be summarized as follows:

- We reveal a shortfall in the previous methods, where simple imitation learning may result in students mimicking the teacher's reasoning style, thus diminishing the versatility of CoTs.
- We are the first attempt to make students learn key reasoning steps from our produced dual CoTs data, further promoting thinking and improving reasoning capabilities.
- Extensive experiments validate the effectiveness of our method across both IND and OOD datasets, showing that MisAiD can create CoTs of superior quality.

[1]We define this as the pivotal moments in reasoning that significantly influence subsequent thought processes.

2 Related Works

CoT Reasoning The emergent ability appears in LLMs across a wide range of NLP tasks (Chowdhery et al., 2023; Wei et al., 2022a). One such ability is CoT reasoning, which involves generating a series of intermediate reasoning steps. While CoT prompting techniques (Wei et al., 2022b) significantly enhance the problem-solving capabilities of models (Kojima et al., 2022; Wang et al., 2023b; Huang et al., 2023), it has little effect on smaller models (Wei et al., 2022a). Chung et al. (2022) suggest that CoT reasoning can be induced in SLMs via instruction tuning on CoTs data. Our work show that the CoT capabilities of SLMs can be further improved by learning from key reasoning steps in dual CoTs data.

Knowledge Distillation from LLMs There has been a lot of work dedicated to distilling knowledge (Hinton et al., 2015) from powerful proprietary LLMs, e.g. ChatGPT (OpenAI, 2023a) in a black-box setting. However, most of these works primarily focus on the general ability distillation by instruction tuning on large and diverse datasets (Taori et al., 2023; Chiang et al., 2023; Peng et al., 2023; Jiang et al., 2023). In contrast, we aim to distill the CoT reasoning capabilities from LLMs same as the standard CoTs distillation (Magister et al., 2023; Ho et al., 2023). Besides, some studies (Li et al., 2022; Hsieh et al., 2023; Liu et al., 2023) employ LLM's rationale or self-evaluation output to enhance SLM's reasoning in a multi-task learning framework. Fu et al. (2023) fine-tune SLMs on four types of reasoning data to ensure out-of-distribution generalization. Wang et al. (2023c) distill SLMs by learning from self-reflection and feedback from LLMs in an interactive multi-round paradigm. Different from the above works, we assist CoTs distillation with teachers' mistakes to alleviate the style imitation of teacher's reasoning.

Learning from Mistakes Recent studies have tried to use mistaken data to enhance the performance of LMs. Shinn et al. (2023) propose Reflexion that allows the LLM agent to self-reflect from its mistakes. Wang and Li (2023) introduces a study assistant that collects and retrieves LLMs' training mistakes to guide future inferences. However, both of the above two methods require the models to be large enough to have basic CoT reasoning or instruction-following capabilities, which is almost impossible to occur in vanilla SLMs. Wang

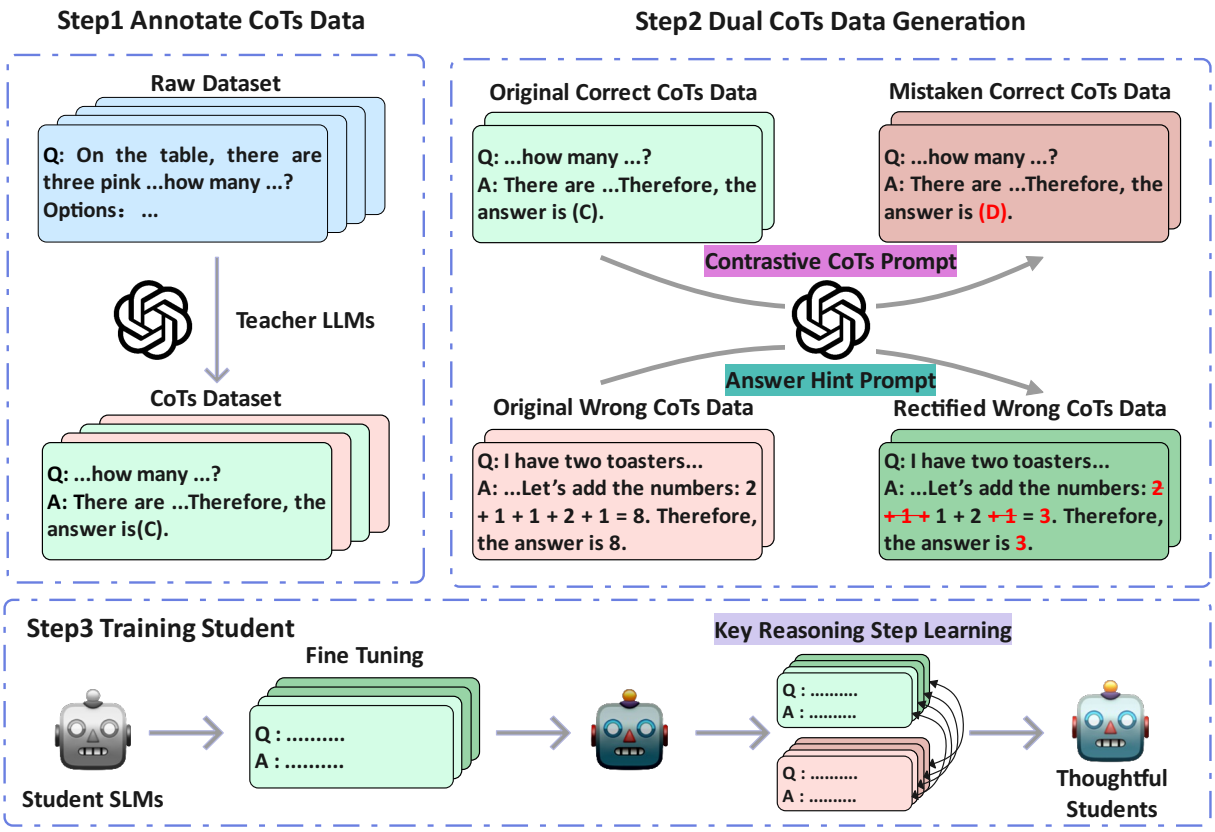

Figure 2: **Overview of our mistake-assisted distillation.** (1) We first retain all CoTs data annotated by teacher LLMs (2) and ask teacher LLMs to generate dual CoTs data using our designed two comprehensive prompts. (3) Then we fine-tune student SLMs on both original correct and rectified-after CoTs data. Finally, we apply key reasoning step learning on the pre-tuned student SLMs by identifying the minor difference between the dual CoTs.

et al. (2023a) propose fine-tuning on counterfactual data to ensure the faithful reasoning of the student model. An et al. (2023) propose LEMA that fine-tunes language models on corrected mistake data, where the mistakes are collected from various LLMs e.g. LLaMA-2-70B (Touvron et al., 2023), WizardLM-70B (Xu et al., 2023), and corrected by GPT-4 (OpenAI, 2023b). In contrast, we collect the teachers' mistakes to create a dual CoTs dataset for further key reasoning steps learning.

3 Methodology

As shown in Figure 2, we propose a mistake-assisted distillation method that makes the student SLM learn the thinking way from the teachers' mistakes, i.e., understanding the reason leading to the teachers' both correct and incorrect inference, rather than merely imitate teacher's reasoning style. Concretely, (1) unlike prior works (Magister et al., 2023; Shridhar et al., 2023; Hsieh et al., 2023; Wang et al., 2023c) that only focus on correct CoTs annotated by teacher LLMs, we first retains all CoTs reasoning data, regardless of its correctness. This strategy aims to prepare the teachers' vanilla CoTs data to understand the reasons behind the teacher's successes and failures in the further steps. (2) To achieve this goal, we construct dual CoTs datasets consisting of positive-negative pairs. Specifically, we design two contextual prompts to instruct teacher LLMs in generating dual CoTs data that follow similar intermediate reasoning steps but lead to divergent conclusions. (3) Finally, we distill the student SLMs by training on the teacher's correct CoTs reasoning data and further key reasoning steps learning on the dual CoTs datasets.

3.1 CoTs Annotated by LLMs

We utilize CoT Prompting (Wei et al., 2022b) to elicit and extract CoTs for a raw dataset $\mathcal{D} = \{(q, a^*)\}$ from LLMs, where q is the question and a^* is the golden answer. Specifically, we first prepare several examples $\mathcal{E} = \{(q_e, z_e)\}$ including question examples q_e and human-curated CoTs z_e, and create a prompt template $\mathcal{T}$ that contains the task description, which can be found in Appendix

C.1. For each $q \in \mathcal{D}$, we extract CoTs as follows:

$$z = (r, a) \sim LLM\left(\mathcal{T} \oplus \mathcal{E} \oplus q\right) \quad (1)$$

where r represent the intermediate reasoning steps and a denotes the final answer in the CoT z. Then, we classify CoTs annotated dataset into two sub-datasets $\mathcal{D}_{IC} = \{(q, z_{ic}, a^*) \mid \forall (q, a^*) \in \mathcal{D}, a \neq a^*\}$ and $\mathcal{D}_{IW} = \{(q, z_{iw}, a^*) \mid \forall (q, a^*) \in \mathcal{D}, a = a^*\}$ that represent the teacher's original correct CoTs dataset and wrong CoTs dataset respectively.

3.2 Dual CoTs Generation

In this subsection, we will introduce how to generate dual CoTs data that follow similar reasoning steps but divergent conclusions for both original correct and wrong CoTs datasets.

Rectify Wrong CoTs Inspired by Rationalization (Zelikman et al., 2022), we design an **Answer Hint Prompt (AHP)** $\mathcal{H}$ that shares the same examples as illustrated in §3.1 but with different organizational structures. The template of AHP can be found in Appendix C.2. Each example in the context and the final provided question in the $\mathcal{H}$ will be inserted with a hint that tells LLMs the answer first before CoTs. Thus, due to the same examples and hint answers, teacher LLM can rectify its original wrong CoTs data with similar reasoning steps but correct answers, which can be expressed as:

$$z_{cc} \sim LLM\left(\mathcal{H} \oplus \mathcal{E} \oplus q \oplus a^*\right) \quad (2)$$

Then we have the corrected wrong CoTs dataset $\mathcal{D}_{CC} = \{(q, z_{cc}, a^*) \mid (q, z_{iw}, a^*) \in \mathcal{D}_{IW}\}$.

Mistake Correct CoTs To generate wrong CoTs data contrasting to the original correct CoTs, a straightforward approach is to use AHP with wrong hint answers to prompt the LLM to produce wrong CoTs. However, in practice, we find that LLMs rarely follow the incorrect answer hints and still generate correct CoTs. We speculate that this may be due to the simplicity of these questions, falling within the knowledge range mastered by LLMs. Additionally, LLMs, having undergone Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), may resist providing unhelpful answers for users. Therefore, we design a **Contrastive CoTs Prompt (CCP)** $\mathcal{C}$ to entice LLMs to generate wrong CoTs, which also leverages the strong in-context-learning (ICL) capabilities of LLMs. The prompt template can be found in Appendix C.3.

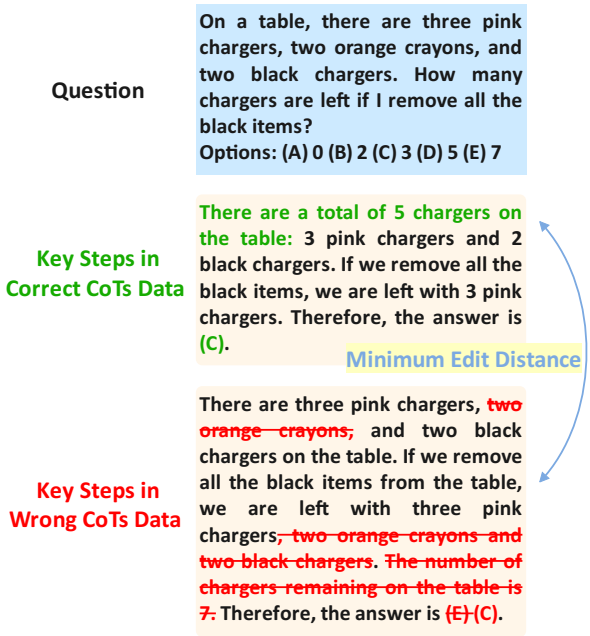

Figure 3: Examples of identifying key reasoning steps in dual CoTs, where the correct CoT and the wrong CoT are dual to each other. The identified key steps in correct reasoning and wrong reasoning are respectively marked in green and red.

To ensure the generation of high-quality wrong CoTs, we randomly sample negative examples from D_{IW} and positive examples from D_{CC}, then curated joint in-context examples $\mathcal{EE} = \{q, z_{iw}, z_{cc}\}$, and extract the wrong CoTs:

$$z_{cw} \sim LLM\left(\mathcal{C} \oplus \mathcal{EE} \oplus q \oplus z_{ic}\right) \quad (3)$$

Then we have the mistaken correct CoTs dataset $\mathcal{D}_{CW} = \{(q, z_{cw}, a^*) \mid (q, z_{ic}, a^*) \in \mathcal{D}_{IC}\}$.

3.3 Training Student with CoTs

Fine-tuning on Correct CoTs After preparing the dual CoTs, we first fine-tune student models on the teachers' original correct CoTs dataset $\mathcal{D}_{IC}$ and rectified wrong reasoning dataset $\mathcal{D}_{CC}$. The training objective is as follows[2]:

$$\pi_{sft} = \arg\max_{\pi} \mathbb{E}_{q,z\sim\mathcal{D}_C}\left[\log \pi(z \mid q)\right] \quad (4)$$

where the merged correct CoTs dataset $\mathcal{D}_C = \mathcal{D}_{IC} \cup \mathcal{D}_{CC}$, and π_{sft} denotes the student with the base inference ability after the initial fine-tuning.

Key Reasoning Steps Learning Inspired by Guo et al. (2023a) who leverage fine-grained quality signals to align human preference, we propose a key

[2]For simplicity, in the following section, we will not formally present a^* in D.

reasoning steps learning (KRSL) method to further encourage students to comprehend the reasons behind both correct and wrong CoTs from the teacher. The specific approach is as follows:

(1) We pair the teacher's original correct CoTs dataset $\mathcal{D}_{IC}$ with its mistaken dataset $\mathcal{D}_{CW}$, creating a originally correct dual CoTs dataset $\mathcal{D}_{C-dual} = \{(q, z_{ic}, z_{iw})\}$, where z_{ic} and z_{iw} are dual to each other. Similarly, we construct the inherently wrong dual CoTs dataset $\mathcal{D}_{W-dual} = \{(q, z_{cc}, z_{cw})\}$. Finally, by merging the two dual datasets, we obtain the ultimate dual CoTs datasets $\mathcal{D}_{dual} = \mathcal{D}_{C-dual} \cup \mathcal{D}_{W-dual}$, which is employed for the subsequent learning of key reasoning steps.

(2) Then we employ the minimum edit distance to identify the key steps in both correct reasoning and wrong reasoning, as shown in Figure 3. In this way, students can identify text segments that are added or replaced in wrong CoTs compared to correct CoTs, and vice versa. These text segments are considered key reasoning steps. After that, we assign token-level weights to facilitate fine-grained learning for correct CoTs z_c and wrong CoTs z_w in D_{dual} respectively:

$$\omega_{c,t} = \begin{cases} \alpha, & \text{if } z_{c,t} \text{ is inserted or replaced} \\ 0, & \text{otherwise} \end{cases},$$
$$\omega_{w,t} = \begin{cases} \beta, & \text{if } z_{w,t} \text{ is deleted or replaced} \\ 0, & \text{otherwise} \end{cases}. \quad (5)$$

where $\alpha \geq 0$, $\beta \geq 0$ and $\omega_{c,t}$ represents the weight of t-th token in z_c (semantically same with $\omega_{w,t}$).

(3) Finally, to ensure that the student makes correct decisions on key steps in correct reasoning, we optimize the student model on these tokens with weighted negative log likelihood. Conversely, to prevent the student from making key steps present in wrong reasoning, we optimize the student model on these steps with weighted positive log likelihood. The sum of both is taken as the final loss. The optimization objective is as follows:

$$\max_{\pi_{sft}} \mathbb{E}_{q,z_c,z_w \sim \mathcal{D}_{dual}} \left[\mathcal{L}(\pi_{sft}, q, z_c, \omega_c) - \mathcal{L}(\pi_{sft}, q, z_w, \omega_w)\right] \quad (6)$$

where

$$\mathcal{L}(\pi, q, z, \omega) = -\sum_{z_t \in z} \omega_t \log \pi(z_t \mid q, z_{<t}) \quad (7)$$

4 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of MisAiD on 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) consists 27 challenging tasks that span arithmetic, symbolic reasoning et al. This collection is maly composed of multiple-choice questions, alongside a minority of open-ended questions. To underscore the superiority of our method, we divide the BBH dataset into two parts for each subtask: a training set (BBH-train) for distillation and a test set (BBH-test) for in-domain evaluation, following a 4:1 ratio.

4.1.2 Out-of-domain

BIG-Bench Sub (BB-sub) is derived from the BIG-Bench (BB) (Guo et al., 2023b), which includes 203 tasks covering linguistics, mathematics, common-sense reasoning, and more. To simplify our evaluation, we refine the selection of tasks from BB by identifying those associated with keywords such as "multiple-choice" and "reasoning."[3] Additionally, we exclude any tasks that are part of the BBH dataset, effectively narrowing down our pool to 61 distinct subtasks. For each of these subtasks, we randomly chose up to 100 instances, culminating in the compilation of the BB-sub dataset.

AGIEval (Zhong et al., 2023) is a benchmark that assesses LMs on reasoning capabilities using human exams across various fields, including English, Math, Law, and Logic. We focused on the English multiple-choice questions within this benchmark to evaluate our method's effectiveness.

AI2 Reasoning Challenge (ARC) (Clark et al., 2018) comprises ARC-Easy (ARC-E) and ARC-Challenge (ARC-C), each catering to different levels of difficulty from middle and high school science exams. ARC-E features simpler questions, whereas ARC-C includes more advanced and challenging ones. We use their test set for our evaluation. Detailed statistics for all mentioned benchmark are provided in the Appendix B.1.

4.2 Models & Baselines & Setup

Models We employ the modern and widely-used open-source language model, LLaMA2-7B (Touvron et al., 2023), as our student SLM. For the teacher model, given its performance

[3] https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md.

Method	Distill?	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
Few-shot	×	15.1	28.5	25.5	25.5	25.4	24.0
Zero-shot-CoT	×	10.6	7.7	7.1	18.4	14.8	11.7
Few-shot-CoT	×	16.3	25.3	9.9	17.2	17.2	17.2
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
MisAiD (ours)	✓	**60.9**$_{+4.1}$	**31.1**$_{+0.8}$	**25.9**$_{+3.9}$	**64.1**$_{+4.5}$	**50.5**$_{+5.4}$	**46.5**$_{+4.7}$
w/o HDA	✓	55.1$_{-1.7}$	30.1$_{-0.2}$	24.1$_{+2.1}$	60.3$_{+0.7}$	44.1$_{-1.0}$	42.7$_{+0.9}$
w/o KRSL	✓	59.7$_{+2.9}$	30.0$_{-0.3}$	24.5$_{+2.5}$	61.9$_{+2.3}$	45.5$_{+0.4}$	44.3$_{+2.5}$

Table 1: Results (Accuracy, %) of the main experiment. w/o HDA represents that fine-tuning student models only on the teacher's original correct CoTs dataset D_{IC} in MisAiD and w/o KRSL denotes that fine-tuning student models on merged correct CoTs dataset D_C without further key reasoning steps learning. The improvements of MisAiD and its variants, w/o HDA and w/o KRSL, over the best baseline in each dataset are indicated by subscripts.

and cost-effectiveness, we employ OpenAI's advanced black-box LLM, ChatGPT, specifically using the "`gpt-3.5-turbo-0613`" variant for extracting CoTs with the same manual prompt that is used in (Suzgun et al., 2023).

Baselines To demonstrate the effectiveness of our proposed method, we compare it with the following baselines: (1) **Teacher & Vanilla Student** under various settings, e.g., Zero-shot (+ CoT) or Few-shot (+ CoT). (2) **Std-CoT** (Magister et al., 2023), which is a standard CoTs distillation method that directly fine-tunes student SLMs on CoTs data. (3) **MT-CoT** (Li et al., 2022) is a multi-task CoTs distillation strategy that aims to optimize both the prediction of answers and the learning of CoTs concurrently. (4) **SCOTT** aims to bolster the reasoning consistency in the student SLMs by integrating counterfactual data into its training regimen.

Setup We employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning of the student SLMs. We empirically set α in KRSL as 1.0 and β as 0.025. Our experiments leverage a mixed-precision training strategy, carried out on 4 $\times$ A100 GPUs. We employ vLLM[4] (Kwon et al., 2023) to enhance inference speed, using a greedy decoding method for text generation on a single A100 GPU. More training details and hyperparameter settings can be found in Appendix B.2.

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

4.3 Main Results

The main experimental results are presented in Table 1, where we compare our proposed method MisAiD with the baselines across both IND and OOD datasets. We aim to illustrate the results by answering the following research questions.

Can CoT distillation improve the performance of students? From the table, it is evident that the student SLMs with distillation outperform those that were not distilled. This demonstrates that the reasoning ability of LLMs can be effectively transferred to SLMs by distilling CoTs.

Can MisAiD further enhance the performance of students compared to other distillation methods? It can be observed that our proposed method MisAiD outperforms the distillation baselines on both IND and OOD datasets, achieving an average improvement of 4.7 % compared to the standard CoT distillation (Std-CoT), which demonstrates the effectiveness and generalizability of MisAiD.

How significant are the improvements in MisAiD attributed to the rectified wrong CoTs and the key steps learning, respectively? Ablation

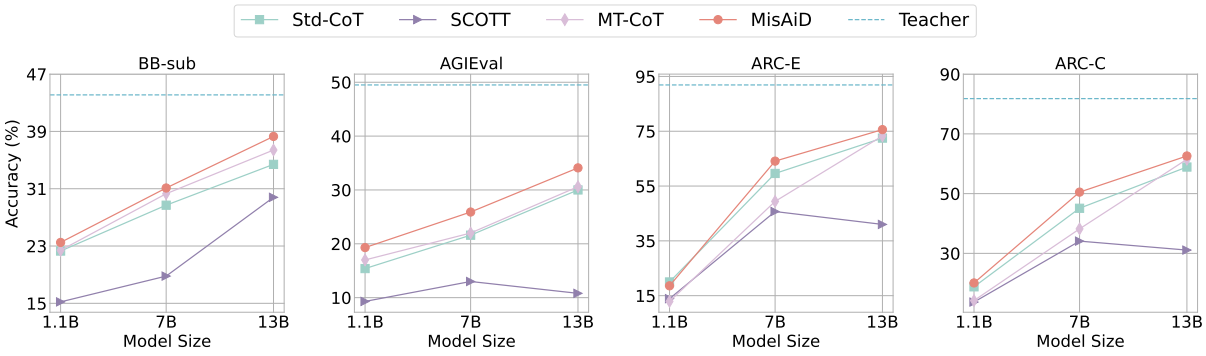

Figure 4: Ablation results on model size for four OOD datasets. The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting. Due to the space limitation, we present the results on the IND dataset in Appendix A.

results in the table show that removing the rectified wrong CoTs (w/o HDA) and removing key reasoning steps learning (w/o KRSL) result in performance degradation on almost all IND and OOD, emphasizing the importance of both components. On the one hand, the rectified teachers' mistakes aid the students in learning diverse ways of thinking. On the other hand, KRSL directs the student's attention to crucial steps in the dual CoTs, thereby improving the reasoning ability of the students.

4.4 Ablation on Model Sizes and Data

In this subsection, we carry out further ablation studies focusing on the sizes of student models and the data employed in KRSL (Key Reasoning Step Learning) to assess the scalability and robustness of MisAiD.

MisAiD is universally applicable to SLMs with various sizes. To better adapt to the community's varying computational resource requirements, we conducted experiments on models of different sizes, including TinyLLaMA-1.1B[5] (Zhang et al., 2024), LLaMA2-7B, and LLaMA2-13B. The results, depicted in Figure 4, demonstrate that MisAiD outperforms the baselines across different model sizes. Particularly on benchmarks with broader evaluation dimensions such as BB-sub and AGIEval, significant improvements are observed regardless of the model size. This suggests that the more challenging a task is, the more it requires genuine reasoning rather than mere imitation, highlighting the benefits that MisAiD brings to student SLMs.

Correct key reasoning steps have a greater impact than incorrect ones. We conduct an ablation study on the key reasoning steps in KRSL (refered to subsection 3.3), where students learn exclusively from either the correct or wrong reasoning steps. The results shown in Figure 5 indicate that learning key reasoning steps solely from either correct or wrong CoTs leads to a decline in performance. This demonstrates that joint learning from both correct and wrong key reasoning steps is more beneficial for enhancing students' reasoning capabilities. Furthermore, we observe a greater performance drop in the absence of key steps in correct CoTs (w/o Correct) compared to the absence of key steps in wrong CoTs (w/o Wrong), suggesting that key steps from correct CoTs have a more significant impact on students' learning.

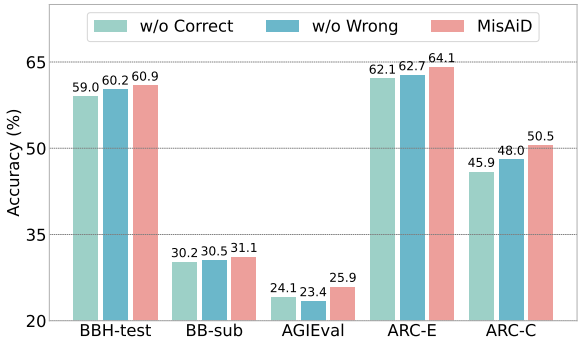

Figure 5: Ablation results on key reasoning steps for the IND (BBH-test) and OOD (others) datasets. w/o Correct represents that students only learn key reasoning steps in wrong CoTs and w/o Wrong represents that students only learn key reasoning steps in correct CoTs.

The quality of dual CoTs data is more important than quantity. We also explore which component of the dual CoTs dataset in KRSL plays a more significant role: the originally correct dual CoTs $\mathcal{D}_{C-dual}$ or the inherently wrong dual CoTs

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

$\mathcal{D}_{W-dual}$. From the Table 2, compared to using $\mathcal{D}_{C-dual}$, employing $\mathcal{D}_{W-dual}$ resulted in superior performance, even with less data, which demonstrates that $\mathcal{D}_{W-dual}$ has higher data quality compared to $\mathcal{D}_{C-dual}$. The dual CoTs constructed from the inherent wrong CoTs of teachers more effectively highlights the key steps in reasoning.

Dataset	$\mathcal{D}_{C-dual}$ (# = 3805)	$\mathcal{D}_{W-dual}$ (# = 1402)	$\mathcal{D}_{dual}$ (# = 5207)
BBH-test	61.3	60.9	60.9
BB-sub	31.2	30.8	31.1
AGIEval	24.4	26.0	25.9
ARC-E	64.6	63.8	64.1
ARC-C	48.9	50.5	50.5
AVG	46.1	46.4	46.5

Table 2: Performance (Accuracy, %) comparison across dual CoTs datasets used in KRSL. The $\mathcal{D}_{C-dual}$ and $\mathcal{D}_{W-dual}$ represents that only the originally correct dual CoTs dataset or the inherently wrong dual CoTs dataset is used in KRSL.The $\mathcal{D}_{dual}$ represents the merge set of above two dual CoT datasets. # denotes the size of the dual CoT datasets.

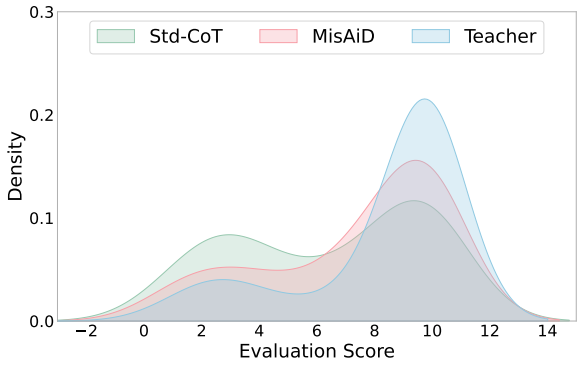

Figure 6: Score distribution evaluated by GPT-4 on BBH-test. We use kernel density estimation to visualize the distribution of CoTs quality scores.

5 Analysis

5.1 Quality of Generated CoTs

Beyond accuracy in reasoning, the quality of CoTS is crucial for interpretable AI. Therefore, we leveraged the sota LLM, GPT-4, to score the quality of CoT generated by Std-CoT, MisAiD, and teacher LLMs. The evaluation focused on which CoT best reflects the key reasoning steps in the problem-solving process, with the specific prompt template detailed in the Appendix C.4. The distribution of the evaluation scores is shown in Figure 6, where we observe that the score distribution for CoTs generated by MisAiD is closer to that of the teacher's compared to Std-CoT. This illustrates that our proposed method is more effective in learning the teacher's way of thinking, resulting in the production of high-quality reasoning.

5.2 Case Study

To more clearly demonstrate the quality of generated CoTs, we present 5 cases sampled from BBH, AGIEval, and ARC, compared with Std-CoT and teachers, as detailed in the Appendix D. Tables 13 and 14 show that the reasoning style of the student SLMs distilled by Std-CoT is very similar to that of the teacher. However, the student SLMs distilled by MisAiD exhibits a changed way of thinking, leading to the correct answers. Table 15 reveals nearly identical reasoning among the three, yet in the critical reasoning steps 7 and 8, Std-CoT fails to make the correct decisions, whereas MisAiD correctly executes stack operations. Cases from OOD datasets, shown in Tables 16 and 17, indicate that MisAiD is capable of accurately analyzing problems and providing more logical reasoning.

6 Conclusion

In this paper, we propose a novel mistake-assisted distillation method to alleviate student imitation of teachers' reasoning styles. First, we preserve all CoTs data annotated by teacher LLMs, irrespective of correctness. Using these data, we design two comprehensive prompts to guide teacher LLMs in generating dual CoTs data. Finally, we utilize the minimum edit distance algorithm to identify the key reasoning steps and employ a fine-grained loss function for guided learning. Extensive experiments demonstrate MisAiD's effectiveness in enhancing student SLMs' reasoning capabilities, outperforming baseline methods on both in-domain and out-of-domain benchmark datasets. We hope our work can make the community attach the importance of learning key reasoning steps in dual CoTs, collectively advancing the efficiency of CoT reasoning distillation.

Limitations

In this section, we discuss the limitations of our study while also offering potentially useful suggestions for future research.

1. Due to the considerations of costs, such as API calls and GPU training expenses, we only use

ChatGPT as the teacher LLM and the widely available open-source model LLaMA2 as the student. Employing GPT-4 as the teacher to provide high-quality examples of annotated CoTs and dual CoTs could better demonstrate the effectiveness of our proposed method.

2. As the dataset sizes and sequence lengths increase, the time required to compute the minimum edit distance in KRSL grows, even with the use of dynamic programming algorithms. There is a future need to explore efficient methods for calculating edit distances in long sequences to address this computational challenge.

3. Currently, most assessments of CoT distillation focus primarily on accuracy (Magister et al., 2023; Ho et al., 2023; Shridhar et al., 2023; Wang et al., 2023c), which is insufficient because safe LLMs rely heavily on trustworthy CoTs. We hope the community to develop standards for evaluating the quality of CoTs, rather than relying solely on automatic assessments by GPT-4.

Ethics Statement

Although the raw dataset used in this work is open source, our approach employs ChatGPT-annotated CoTs data to distill knowledge into a student model. This process may inadvertently pass on the societal biases (Schaeffer et al., 2023) and hallucination (Zhang et al., 2023) issues inherent in LLMs. To address this, we propose MisAiD that extends beyond mere fine-tuning on teacher-annotated data. By adopting a learning strategy focused on critical reasoning steps from teachers' mistakes, our method aims to reduce the imitation of the teacher's reasoning style and encourages independent thought. This approach helps mitigate the problem of inheriting biases to a certain extent.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes LLM better reasoner. *CoRR*, abs/2310.20689.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

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.

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. 2023a. Beyond imitation: Leveraging fine-grained quality signals for alignment. *CoRR*, abs/2311.04072.

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

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

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. 2022. 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.

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

OpenAI. 2023b. Gpt-4 technical report. `https://cdn.openai.com/papers/gpt-4.pdf`. Accessed: [insert date here].

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

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. 2023. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *CoRR*, abs/2303.11366.

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.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10667–10685.

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*.

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.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*.

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 Model Size for In-domain Dataset

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

B Details of Experiment

B.1 Dataset Statistics

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

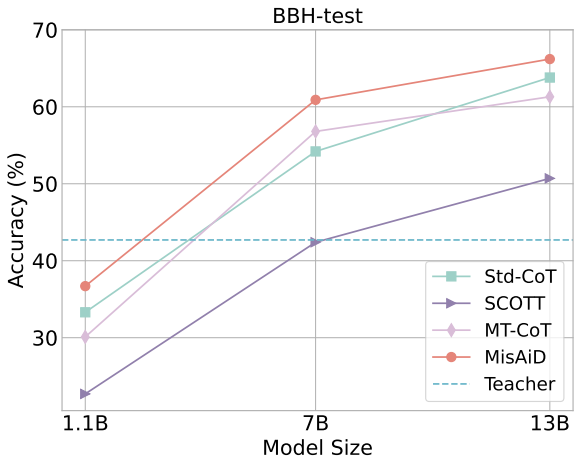

Figure 7: Ablation study on model size for the IND dataset (BBH-test). The dotted line indicates the performance of the teacher LLM under the Zero-shot-CoT setting.

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.

B.2 Hyperparameters Settings

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

C Prompt Templates

C.1 Prompt 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 Answer Hint Prompt

We list the Answer Hint Prompt templates in Table 10, which imply the teacher LLMs to generate the CoTs based on the given answers following the in-context examples.

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.

C.3 Contrastive CoTs Prompt

We list the Contrastive CoTs Prompt templates in Table 11, which query the teacher LLMs to generate the CoTs with similar rationales to the original ones but divergent answers by following the few examples provided with contrastive CoT pairs.

C.4 Evaluation Prompt of CoTs Quality

We list the evaluation prompt templates of CoTs quality in Table 12.

D Case Study

Here we show 5 cases in Table 13, 14, 15, 16 and 17 to clearly compare the CoT generated by MisAiD with the teacher LLM and the standard CoTs distillation (Std-CoT). We utilize ✓ and ✗ to denote whether the CoT is correct or incorrect, 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 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.2}
A: Let's think step by step. {Human-Curated-CoTs No.3}.

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

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

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

Q: {Task Example Question No.1}
H: {The correct answer is [HINT ANSWER No.1]}
A: Let's think step by step. {Human-Curated-CoTs No.1}.

Q: {Task Example Question No.2}
H: {The correct answer is [HINT ANSWER No.2]}
A: Let's think step by step. {Human-Curated-CoTs No.2}.

Q: {Task Example Question No.3}
H: {The correct answer is [HINT ANSWER No.3]}
A: Let's think step by step. {Human-Curated-CoTs No.3}.

Q: {QUESTION}
H: {The correct answer is [HINT ANSWER]}
A: Let's think step by step.
```

Table 10: Answer Hint Prompt templates for rectifying the wrong CoTs data based on the hint answers.

```
{Task Description}. You need to complete the [Wrong Response] which requires you to give the
most likely incorrect answer to the [Question] and the rationale for the incorrect answer.
The incorrect answer and rationale in the [Wrong Response] must be different from the correct
answer and rationale in the [Right Response].

[Question]: {Task Example Question No.1}
[Right Response]: {Corrected CoT No.1}
[Wrong Response]: {Wrong CoT No.1}

[Question]: {Task Example Question No.2}
[Right Response]: {Corrected CoT No.2}
[Wrong Response]: {Wrong CoT No.2}

[Question]: {Task Example Question No.3}
[Right Response]: {Corrected CoT No.3}
[Wrong Response]: {Wrong CoT No.3}

[Question]: {USER_QUESTION}
[Right Response]: {Corrected CoT}
[Wrong Response]:
```

Table 11: Contrastive CoT Prompt templates for mistaken the correct CoTs data. The examples are sampled from the teachers' original wrong CoTs data and its corrected CoTs. In this way, teacher LLMs can expose the reasoning flaws in problems that were originally solved correctly.

```
[System] You are a helpful and precise assistant for assessing the quality of the response.

[Question]: {QUESTION}
[Reference Answer]: {ANSWER}

[AI Assistant 1's Answer Start]
{ASSISTANT1}
[AI Assistant 1's Answer End]

[AI Assistant 2's Answer Start]
{ASSISTANT2}
[AI Assistant 2's Answer End]

[AI Assistant 3's Answer Start]
{ASSISTANT3}
[AI Assistant 3's Answer End]

[System] We would like to request your feedback, in the form of scoring, on which of the
responses from AI Assistant 1, 2 and 3 effectively demonstrates the key reasoning steps in
solving this question. Key Reasoning Steps refer to certain crucial steps in the process of
logical reasoning or problem-solving. These steps play a significant role in the thinking
process and have a notable impact on subsequent reasoning. Each student will receive an
overall score on a scale of 1 to 10, where a higher score signifies that the assistant's
response is more effectively demonstrates the key reasoning steps for the question.
Please provide a comprehensive explanation, avoiding any potential bias and ensuring that
the order in which the responses were presented does not affect your judgment. And then
output three lines indicating the scores for AI Assistant 1, 2 and 3, respectively.

Output with the following format:
Evaluation evidence: <your evaluation explanation here>
Score of AI Assistant 1: <score>
Score of AI Assistant 2: <score>
Score of AI Assistant 3: <score>
```

Table 12: Prompt template of GPT-4 for assessing CoTs quality. In the analysis, we use this template to eval the quality of CoTs generated by Std-CoT, MisAiD and the teacher LLM respectively.

Question:	On the table, there are three pink scrunchiephone chargers, two orange crayons, and two black scrunchiephone chargers. If I remove all the black items from the table, how many scrunchiephone chargers remain on it? Options:(A) zero(B) one (C) two (D) three (E) four (F) five (G) six (H) seven (I) eight (J) nine (K) ten (L) eleven (M) twelve (N) thirteen (O) fourteen (P) fifteen (Q) sixteen
Teacher	According to this question, there are three pink scrunchiephone chargers, two orange crayons, and two black scrunchiephone chargers on the table. If we remove all the black items from the table, we are left with three pink scrunchiephone chargers and two orange crayons. Therefore, the number of scrunchiephone chargers that remain on the table is five. Therefore, the answer is (F). ✗
Std-CoT	According to this question, there are three pink scrunchiephone chargers, two orange crayons, and two black scrunchiephone chargers on the table. If we remove all the black items from the table, we are left with three pink scrunchiephone chargers, two orange crayons, and two black scrunchiephone chargers. The number of scrunchiephone chargers remaining on the table is seven. Therefore, the answer is (H). ✗
MisAiD	According to this question, there are a total of 5 scrunchiephone chargers on the table: 3 pink scrunchiephone chargers and 2 black scrunchiephone chargers. If we remove all the black items, we are left with 3 pink scrunchiephone chargers. Therefore, the answer is (D). ✓

Table 13: A reasoning about colored objects case from BIG-Bench Hard.

Question:	Find a movie similar to Jurassic Park, Independence Day, Forrest Gump, Batman. Options: (A) The Firm (B) Roger Dodger (C) Innocent Man (D) An (E) The Silence of the Lambs
Teacher	- Jurassic Park (action, adventure, science-fiction; 1993) - Independence Day (action, science-fiction, drama; 1996) - Forrest Gump (comedy, drama, romance; 1994) - Batman (action, adventure; 1989) These are all famous Hollywood movies produced in the 1980s and 1990s. Amongst all the options, the only movie similar to these ones seems to be The Firm (drama, mystery, thriller; 1993), as it is also a popular movie from the same era. Therefore, the answer is (A). ✗
Std-CoT	Jurassic Park is an action and adventure movie. Independence Day is an action, science-fiction, and adventure movie. Forrest Gump is a comedy, drama, and romance movie. Batman is an action, adventure, and crime movie. The Firm is a crime, drama, and thriller movie. Roger Dodger is a comedy and drama movie. Innocent Man is a crime, drama, and thriller movie. The Silence of the Lambs is a crime, drama, and thriller movie. Amongst all the options, the movie similar to Jurassic Park seems to be The Firm. Therefore, the answer is (A). ✗
MisAiD	Issue 1: Is this movie similar to Jurassic Park? Above all the options, the only movie similar to Jurassic Park seems to be The Silence of the Lambs due to its suspenseful and thrilling nature. Issue 2: Is this movie similar to Independence Day? Above all the options, the only movie similar to Independence Day seems to be Roger Dodger due to its sci-fi and action-packed nature. Issue 3: Is this movie similar to Forrest Gump? Above all the options, the only movie similar to Forrest Gump seems to be The Firm due to its drama and emotional depth. Issue 4: Is this movie similar to Batman? Above all the options, the only movie similar to Batman seems to be An due to its superhero and action-packed nature. Based on the similarities outlined in Issues 1, 2, 3, and 4, the movie similar to these given movies seems to be The Silence of the Lambs. Therefore, the answer is (E). ✓

Table 14: A movie recommendation case from BIG-Bench Hard.

Question:	Correctly close a Dyck-n word. Complete the rest of the sequence, making sure that the parentheses are closed properly. Input: < ((< >)

Teacher	We should process each input one by one and keep track of the stack configuration.
	0: empty stack
	1: < ; stack: <
	2: (; stack: < (
	3: (; stack: < ((
	4: { ; stack: < (({
	5: < ; stack: < (({ <
	6: > ; stack: < (({
	7: } ; stack: < (({
	8:) ; stack: < ((
	Now, we have reached the end. The final stack is "< ((".
	We will need to pop out "(", "(" one by one in that order.
	So, we need ")", ")". Therefore, the answer is)). ✗

Std-CoT	We should process each input one by one and keep track of the stack configuration.
	0: empty stack
	1: < ; stack: <
	2: (; stack: < (
	3: (; stack: < ((
	4: { ; stack: < (({
	5: < ; stack: < (({ <
	6: > ; stack: < (({
	7: } ; stack: < (({
	8:) ; stack: < (({
	Now, we have reached the end. The final stack is "< (({".
	We will need to pop out "{", "(", "<" one by one in that order.
	So, we need "}", ")", ">". Therefore, the answer is }) >. ✗

MisAiD	We should process each input one by one and keep track of the stack configuration.
	0: empty stack
	1: < ; stack: <
	2: (; stack: < (
	3: (; stack: < ((
	4: { ; stack: < (({
	5: < ; stack: < (({ <
	6: > ; stack: < (({
	7: } ; stack: < ((
	8:) ; stack: < (
	Now, we have reached the end. The final stack is "< (".
	We will need to pop out "(", "<" one by one in that order.
	So, we need ")", ">". Therefore, the answer is) >. ✓

Table 15: A dyck languages case from BIG-Bench Hard.

Question:	Anna has 4 books. If she decide to arrange the 4 books in every possible combination and moved just one book every minute, how long would it taken by her ? Options: (A)22 minutes (B)30 minutes (C)15 minutes (D)24 minutes (E)35 minutes
Teacher	First, Anna has 4 books and she wants to arrange them in every possible combination. To find the number of possible combinations, we can use the formula for permutations, which is n! (n factorial), where n is the number of items. In this case, n = 4, so the number of possible combinations is 4! = 4 x 3 x 2 x 1 = 24. Next, we know that Anna can move just one book every minute. Since there are 24 possible combinations, it will take her 24 minutes to go through all of them. Therefore, the answer is (D) 24 minutes. ✓
Std-CoT	This question focuses on the time it takes Anna to arrange the 4 books in every possible combination. We know that Anna moved just one book every minute. Plugging in the minutes per move: 1 minute There are 4 books: A, B, C, D There are (4!) = 4 different combinations: A, B, C, D; B, A, C, D; B, C, A, D; C, A, B, D; D, A, B, C The total number of minutes it would take Anna to arrange the books in every possible combination is the sum of the minutes needed for each combination: (4 minutes) + (4 minutes) + (4 minutes) + (4 minutes) + (4 minutes) = 15 minutes. Therefore, the answer is (C) 15 minutes. ✗
MisAiD	This question asks for the total amount of time it would take Anna to arrange the books in every possible combination and moved just one book every minute. We first need to calculate the number of possible combinations: 4! = 24. Then, we need to multiply the number of possible combinations by the number of minutes per combination: 24 * 1 minute = 24 minutes. Therefore, the answer is (D) 24 minutes. ✓

Table 16: A AQuA-RAT case from AGIEval.

Question:	Which method is the safest way to watch an eclipse of the Sun? Options: (A) Turn away after two or three minutes. (B) Look at the Sun through a long telescope. (C) Cast an image through a pinhole onto a screen. (D) Blink often until your eyes get used to the light.
Teacher	First, it is important to note that directly looking at the Sun during an eclipse can be extremely dangerous and can cause permanent damage to the eyes. Option (A) suggests turning away after two or three minutes. However, even a brief exposure to the Sun during an eclipse can be harmful, so this method is not safe. Option (B) suggests looking at the Sun through a long telescope. This method is also unsafe because the intense light from the Sun can damage the telescope and cause harm to the eyes. Option (C) suggests casting an image through a pinhole onto a screen. This method is a safe and recommended way to watch an eclipse. By creating a pinhole camera, the image of the Sun is projected onto a screen, allowing for safe viewing without directly looking at the Sun. Option (D) suggests blinking often until your eyes get used to the light. This method is not safe as it does not provide any protection from the harmful rays of the Sun during an eclipse. Therefore, the answer is (C) Cast an image through a pinhole onto a screen. ✓
Std-CoT	We know that looking directly at the Sun can cause permanent damage to the eyes. Therefore, the sanswer is (C) casting an image through a pinhole onto a screen. ✓
MisAiD	We are told that this question is about the safest way to watch an eclipse of the Sun. Looking at the Sun through a long telescope is not recommended as it can cause permanent damage to the eyes. Blinking often until your eyes get used to the light is also not recommended as it can cause eye strain and discomfort. The only remaining option is to cast an image through a pinhole onto a screen. This method involves creating a small hole in a card or paper and projecting the Sun's image onto a surface, such as a wall or a piece of paper. This method is safe as it does not require direct eye contact with the Sun. Therefore, the answer is (C). ✓

Table 17: A case from AI2 Reasoning Challenge.