Mind's Mirror: Distilling Self-Evaluation Capability and Comprehensive Thinking from Large Language Models

Weize Liu[1], **Guocong Li**[1], **Kai Zhang**[2], **Bang Du**[1], **Qiyuan Chen**[1],
Xuming Hu[3]*, **Hongxia Xu**[2,4]*, **Jintai Chen**[5], **Jian Wu**[2,4]
[1]Zhejiang University [2]School of Public Health, Zhejiang University
[3]The Hong Kong University of Science and Technology (Guangzhou)
[4]Liangzhu Laboratory and Institute of Wenzhou, Zhejiang University
[5]Computer Science Department, University of Illinois Urbana-Champaign
{weizeliu1115,xuminghu97}@gmail.com einstein@zju.edu.cn

Abstract

Large language models (LLMs) have achieved remarkable advancements in natural language processing. However, the massive scale and computational demands of these models present formidable challenges when considering their practical deployment in resource-constrained environments. While techniques such as chain-of-thought (CoT) distillation have displayed promise in distilling LLMs into small language models (SLMs), there is a risk that distilled SLMs may still inherit flawed reasoning and hallucinations from LLMs. To address these issues, we propose a twofold methodology: First, we introduce a novel method for distilling the self-evaluation capability from LLMs into SLMs, aiming to mitigate the adverse effects of flawed reasoning and hallucinations inherited from LLMs. Second, we advocate for distilling more comprehensive thinking by incorporating multiple distinct CoTs and self-evaluation outputs, to ensure a more thorough and robust knowledge transfer into SLMs. Experiments on three NLP benchmarks demonstrate that our method significantly improves the performance of distilled SLMs, offering a new perspective for developing more effective and efficient SLMs in resource-constrained environments.

1 Introduction

With the gradual increase in the number of parameters, large language models (LLMs) have achieved significant successes in the field of natural language processing (Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2023; OpenAI, 2023). However, the tremendous model sizes and computational requirements of LLMs introduce challenges to their practical application, especially in resource-limited environments (Zhao et al., 2023; Zhu et al., 2023b). To address these challenges, various studies have delved into the compression of LLMs into small language models (SLMs) using knowledge distillation techniques and have led to significant reductions in computational complexity and inference costs (Jiang et al., 2020; Gu et al., 2023; Agarwal et al., 2023). This process involves traditional teacher-student learning methods and the more recent chain-of-thought (CoT) distillation method (Zhu et al., 2023b). The CoT distillation methods use the CoT reasoning process of LLMs as supervision for training SLMs, rather than just labels. This allows SLMs to learn the reasoning process of LLMs, thereby improving the performance of SLMs.

While these CoT distillation methods have proven to be beneficial, they are not without their flaws, particularly:

1. Even during the CoT distillation process, the distilled SLMs remain vulnerable to the flawed supervision provided by LLMs, as observations suggest that chains of thought (CoTs) generated by LLMs may contain hallucinations (Zhang et al., 2023), accumulate errors (Shen et al., 2021), or lack robustness (Vaswani et al., 2017; Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022). As shown in the example in Figure 1, "LLM Random CoT 2" incorrectly broadens the scope of the premise by arguing that "Being an animal welfare advocate means caring about all the animals that inhabit the planet." In practice, it is not easy to exclude these flawed CoTs, since the ground truth of CoTs is not always easily obtainable (Zhang et al., 2023). Training SLMs with these flawed CoTs will result in SLMs inheriting these flaws and performance degradation (Alemohammad et al., 2023; Ho et al., 2023).

2. A single instance of CoT might not capture the diverse reasoning routes LLMs can explore,

*Corresponding authors.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6748–6763
June 16-21, 2024 ©2024 Association for Computational Linguistics

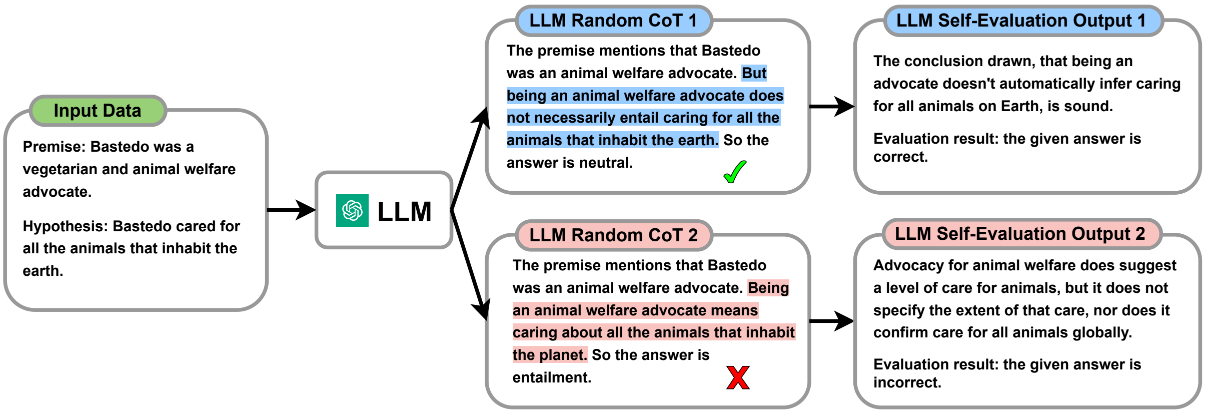
Input Data
Premise: Bastedo was a vegetarian and animal welfare advocate.
Hypothesis: Bastedo cared for all the animals that inhabit the earth.
LLM
LLM Random CoT 1
The premise mentions that Bastedo was an animal welfare advocate. But being an animal welfare advocate does not necessarily entail caring for all the animals that inhabit the earth. So the answer is neutral.
✓
LLM Random CoT 2
The premise mentions that Bastedo was an animal welfare advocate. Being an animal welfare advocate means caring about all the animals that inhabit the planet. So the answer is entailment.
✗
LLM Self-Evaluation Output 1
The conclusion drawn, that being an advocate doesn't automatically infer caring for all animals on Earth, is sound.
Evaluation result: the given answer is correct.
LLM Self-Evaluation Output 2
Advocacy for animal welfare does suggest a level of care for animals, but it does not specify the extent of that care, nor does it confirm care for all animals globally.
Evaluation result: the given answer is incorrect.

Figure 1: Examples of both the random CoT responses and their self-evaluation outputs generated by the LLM during natural language inference tasks. The human-like self-evaluation of the LLM enables the LLM to self-evaluate the correctness of its CoT reasoning processes, identifying which are correct (highlighted in blue) and which are incorrect (highlighted in red) in these randomly generated CoT reasoning.

limiting the richness of the distilled knowledge of SLMs. Furthermore, relying solely on the CoT reasoning process as supervision for training SLMs is insufficient to distill the comprehensive capabilities of LLMs, such as the ability to check the correctness of answers.

To mitigate the impact of these flawed CoTs and allow SLMs to learn more comprehensive capabilities, we propose an innovative methodology that involves training SLMs to possess the self-evaluation capability. Humans often evaluate their reasoning processes to reduce errors in decision-making (Poole and Mackworth, 2010), and a similar self-evaluation capability has also been observed in LLMs (Kadavath et al., 2022; Shinn et al., 2023; Madaan et al., 2024; Paul et al., 2023), which recognizes and corrects the generated hallucinations, faulty reasoning, and harmful content in a CoT (Pan et al., 2023). Figure 1 illustrates this with an example where incorrect reasoning in “LLM Random CoT 2” is identified and corrected in the self-evaluation. The advantage of self-evaluation is that it does not rely on external resources. However, it is constrained by the inherent capabilities of the model. To address this, we guide SLMs in distillation to learn the self-evaluation capability of LLMs. By learning the ability of LLMs to analyze right from wrong, SLMs can understand both what should and should not be generated, enhancing their predictive accuracy and reliability in various NLP tasks.

To facilitate comprehensive thinking and address the randomness and limitations of relying on a single CoT and a single self-evaluation, our second methodology insight involves distilling SLMs from diverse CoTs and multiple self-evaluation outputs generated by LLMs. This enables SLMs to inherit a broader range of comprehensive thinking capabilities since diverse CoTs and self-evaluation collectively offer a more comprehensive perspective, derived from the varied state spaces of LLMs.

In summary, our contributions can be outlined as follows:

1. We distill the self-evaluation capability from LLMs into SLMs, which helps SLMs understand the potential reasons behind correct or incorrect reasoning and lays the foundation for mitigating errors (e.g., hallucinations) arising from flawed CoTs.

2. We distill diverse CoTs and corresponding multiple self-evaluation outputs from LLMs into SLMs, enabling SLMs to learn a more comprehensive state space of LLMs. This approach empowers SLMs with enhanced reasoning and more comprehensive capabilities.

3. Comprehensive experiments demonstrate that our method enables SLMs to inherit the self-evaluation capability and comprehensive thinking of LLMs, significantly enhancing the performance and reliability of distilled SLMs, and outperforming previous CoT distillation methods. This affirms our method is essential for creating robust and efficient SLMs capable of high-quality reasoning in resource-constrained environments.

The code is available at https://github.com/Attention-is-All-I-Need/Mind-s-Mirror-Distilling-LLM.

2 Related Work

Chain-of-thought reasoning Chain-of-thought (CoT) is a prompting method where a model generates intermediate reasoning steps to enhance its problem-solving capabilities (Wei et al., 2022). The chain-of-thought with self-consistency (CoT-SC) (Wang et al., 2023b) builds upon CoT, sampling a set of diverse reasoning paths and selecting the most consistent answer as the final answer. This largely mitigates errors introduced by the inherent randomness of LLMs. The Tree of Thoughts (ToT) method (Yao et al., 2024) models problem-solving as a tree search process, enabling LLMs to explore different reasoning pathways and conduct self-evaluation to determine the solution taken at each step. Therefore, by leveraging the capability of LLMs to generate diverse reasoning paths and self-evaluation, ToT significantly enhances the performance of LLMs in solving tasks such as Game of 24, Creative Writing, and Mini Crosswords.

Self-evaluation in LLMs Many recent works have leveraged the self-evaluation capability of LLMs to enhance the reliability of their responses, such as Self-Refine (Madaan et al., 2024), Self-Check (Miao et al., 2023), SelfCheckGPT (Manakul et al., 2023), and Reflexion (Shinn et al., 2023). Concurrently, other studies have demonstrated the self-improvement potential of LLMs (Huang et al., 2023; Pan et al., 2023), as exemplified by RLAIF (Lee et al., 2023). However, these methods are designed for LLMs and do not consider distilling the self-evaluation capability into SLMs.

Knowledge distillation from LLMs Knowledge distillation enhances the performance of smaller models by transferring knowledge from larger models (Hinton et al., 2015). This method has been widely adopted for the optimization and compression of models. Recent studies have been focusing on leveraging the CoT reasoning generated by LLMs to enhance the performance of SLMs (Wang et al., 2023a; Magister et al., 2023; Shridhar et al., 2023; Wang et al., 2023c; Chen et al., 2023; Fu et al., 2023; Zhu et al., 2023a; Saha et al., 2023). For instance, Hsieh et al. (2023) introduced a "Distilling step-by-step" method for extracting rationales from LLMs as additional supervision for training SLMs. Similarly, Li et al. (2023) proposed the Symbolic Chain-of-Thought Distillation (SCoTD) method, which trains SLMs to learn CoT reasoning. Additionally, Ho et al. (2023) presented "Fine-tune-CoT", a method that generates reasoning samples from LLMs to fine-tune SLMs. However, these methods do not consider mitigating the impact of harmful content in CoTs generated by LLMs on SLMs, as well as distilling other capabilities beyond CoTs. In contrast, our methodology incorporates the self-evaluation capability of LLMs into distillation, which can be utilized to mitigate the effects of flawed CoTs in a completely unsupervised manner and without relying on external resources, and allows SLMs to learn the more comprehensive capabilities of LLMs. Furthermore, some related works utilize SLMs with up to several billion parameters and have not been able to validate their effectiveness on SLMs with as few as 220M parameters, so our approach exhibits lower resource requirements and broader applicability.

3 Distilling Self-Evaluation Capability and Comprehensive Thinking

We propose a new methodology for distilling the self-evaluation capability and comprehensive thinking of an LLM into an SLM. Our overall framework is illustrated in Figure 2, which operates in 4 steps: (1) Given an LLM and an unlabeled dataset, we utilize CoT prompts to generate diverse rationales and corresponding pseudo-labels from the LLM. (2) By devising self-evaluation prompts, we enable the LLM to evaluate the correctness of its generated CoTs, which also include both the rationales and labels in its self-evaluation outputs. (3) Leveraging the rationales and labels in the self-evaluation outputs generated by the LLM, we employ multi-task learning to train the SLM, enabling the SLM to distinguish right from wrong. (4) Utilizing the diverse rationales in CoTs and labels from either LLM-generated pseudo-labels or human-annotated labels, we employ multi-task learning to train the SLM's reasoning capability.

3.1 Obtaining diversity CoTs and self-evaluation outputs from the LLM

In our pipeline, an LLM functions as the teacher, while an SLM serves as the student. First, we let the LLM generate multiple different CoTs and self-evaluation outputs for a given task. We utilize few-shot CoT prompting to enhance the quality and standardize the formats of the CoTs generated by the LLM. This process is shown as step 1 and step 2 in Figure 2.

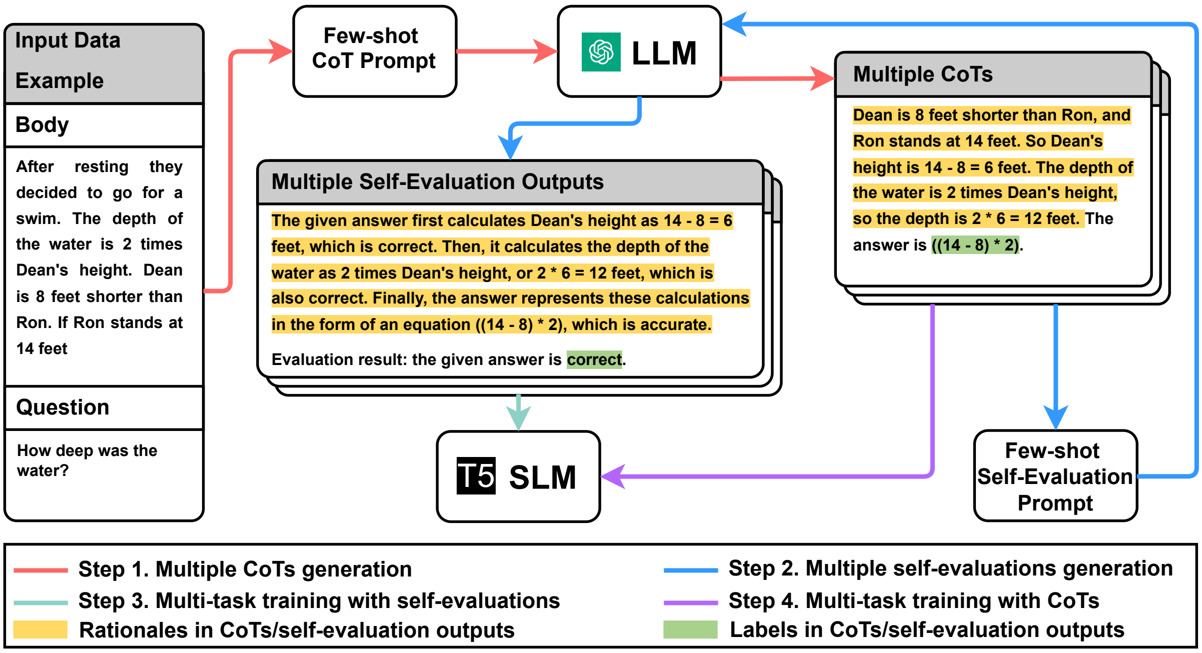
Input Data Example
Body
After resting they decided to go for a swim. The depth of the water is 2 times Dean's height. Dean is 8 feet shorter than Ron. If Ron stands at 14 feet
Question
How deep was the water?
Few-shot CoT Prompt
LLM
Multiple CoTs
Dean is 8 feet shorter than Ron, and Ron stands at 14 feet. So Dean's height is 14 - 8 = 6 feet. The depth of the water is 2 times Dean's height, so the depth is 2 * 6 = 12 feet. The answer is ((14 - 8) * 2).
Multiple Self-Evaluation Outputs
The given answer first calculates Dean's height as 14 - 8 = 6 feet, which is correct. Then, it calculates the depth of the water as 2 times Dean's height, or 2 * 6 = 12 feet, which is also correct. Finally, the answer represents these calculations in the form of an equation ((14 - 8) * 2), which is accurate.
Evaluation result: the given answer is correct.
T5 SLM
Few-shot Self-Evaluation Prompt
Step 1. Multiple CoTs generation
Step 2. Multiple self-evaluations generation
Step 3. Multi-task training with self-evaluations
Step 4. Multi-task training with CoTs
Rationales in CoTs/self-evaluation outputs
Labels in CoTs/self-evaluation outputs

Figure 2: Detailed overview of our proposed methodology. **Step 1**: Obtain multiple CoTs from the LLM (Section 3.1.1). **Step 2**: Obtain multiple self-evaluation outputs from the LLM (Section 3.1.2). **Step 3**: Train the SLM with multiple self-evaluation outputs, enabling the SLM to distinguish right from wrong (Section 3.2.1). **Step 4**: Train the SLM with multiple CoTs to give the SLM comprehensive reasoning capabilities (Section 3.2.2).

3.1.1 Obtaining multiple CoTs

For an unlabeled dataset D, we devise a few-shot CoT prompt template p delineating how the task should be approached. We combine each input data x_i with p and use it as an input to LLM. With examples from p, the LLM can simulate examples to generate the CoT response for x_i that contains a rationale r_i and a pseudo-label y_i (the yellow part and the green part of the "Multiple CoTs Outputs" in Figure 2). We let the LLM regenerate several times to get multiple different CoTs.

3.1.2 Obtaining multiple self-evaluation outputs

After forming multiple CoTs representing different thoughts, a self-evaluation phase is initiated to evaluate the correctness of the CoTs. This is essential to imitate the complete human thought process and correct mistakes in reasoning. Given an unlabeled dataset D, we devise a few-shot self-evaluation prompt template p_{eval}, which guides the LLM in evaluating each CoT's correctness. For each CoT x_c, shown in "Multiple CoTs" in Figure 2, we add it to p_{eval} and use this as an input to prompt the LLM to generate the self-evaluation. With examples in p_{eval}, the LLM simulates examples to generate the self-evaluation output for x_c that also contains a rationale r_{eval_i} and a label y_{eval_i} (the yellow part and the green part of the "Multiple Self-Evaluation Outputs" in Figure 2).

Similarly, to distill a more comprehensive self-evaluation capability of the LLM, we generate multiple different self-evaluation outputs for each CoT. Multiple self-evaluation outputs along with multiple CoTs represent a more comprehensive and complete thought process for the LLM. Additionally, given the randomness of LLM outputs, we suggest examining the quality and diversity of multiple CoTs and self-evaluation outputs generated by the LLM for the same input, and removing duplicates and outputs of inferior quality, to enhance data quality. This is an optional step.

3.2 Training the SLM with multiple self-evaluation outputs and diverse CoTs

After generating diverse CoTs and their corresponding self-evaluation outputs using the LLM, we begin to train the SLM. Our training methodology for SLMs first emphasizes distilling self-evaluation capability to lay the foundation for reducing the impact of errors in CoTs on SLMs, followed by incorporating comprehensive reasoning capability through diverse CoTs distillation. Hsieh et al. (2023) have demonstrated that multi-task learning can lead to better performance than simply treating rationale and label predictions as a single joint task, and can reduce computation overhead during inference since it allows the SLM to directly predict labels without generating rationales. Hence, we employ multi-task learning to train the SLM for self-evaluation capability and CoT reasoning capability. By appending different "task prefixes" at the beginning of the input, we can direct the SLM to generate either a label or a rationale (Raffel et al., 2020). We train the SLM to generate a label when the prefix is "predict: ", and to generate a rationale when the prefix is "explain: ". This process is shown as step 3 and step 4 in Figure 2.

3.2.1 Distilling self-evaluation capability

Using the self-evaluation data generated by the LLM, we aim to distill this capability into the SLM. During this phase, the model is trained to predict the self-evaluation label y_{eval_i} as well as generate corresponding rationale r_{eval_i}. To guide the SLM in learning the self-evaluation outputs for each CoT, we employ a multi-task loss function:

$$L_{SE} = \frac{1}{N_{eval}} \sum_{c=1}^{N_{eval}} \Big(\lambda \ell(f(x_c), y_{eval_c}) + (1-\lambda)\ell(f(x_c), r_{eval_c}) \Big),$$

where f represents the SLM and ℓ is the cross-entropy loss between the tokens predicted by the SLM and the target tokens. x_c is the CoT that needs to be evaluated. λ is a hyperparameter for weighing the rationale loss. y_{eval_c} indicates the self-evaluation label generated by the LLM, r_{eval_c} is the rationale in the c^{th} self-evaluation output, and N_{eval} is the total amount of self-evaluation outputs.

3.2.2 Distilling CoT reasoning capability

After successfully distilling self-evaluation capability, the focus shifts to leveraging diverse CoTs to train the comprehensive reasoning capability of SLMs. For each instance in the dataset, we also employ a multi-task loss function to guide the SLM in learning CoT reasoning by:

$$L_{CoT} = \frac{1}{N_{CoT}} \sum_{i=1}^{N_{CoT}} \Big(\lambda \ell(f(x_i), \hat{y}_i) + (1-\lambda)\ell(f(x_i), r_{CoT_i}) \Big),$$

where x_i indicates input data, $\hat{y}_i$ indicates the pseudo-label y_i generated by the LLM or human-annotated label, r_{CoT_i} is the rationale in the i^{th} CoT, and N_{CoT} is the total amount of CoTs.

This two-pronged training regimen ensures that the SLM is not merely parroting the CoT reasoning but deeply understands introspective self-evaluation and nuanced reasoning, mirroring the powerful cognitive capabilities of the LLM.

4 Experiments

Tasks and datasets To evaluate our distillation method, we conduct comprehensive experiments on three tasks: 1) math word problems (MWPs) task with the SVMAP dataset (Patel et al., 2021); 2) commonsense question answering (CQA) task with the CQA dataset (Talmor et al., 2019; Rajani et al., 2019); 3) natural language inference (NLI) task with the ANLI dataset (Nie et al., 2020). For dataset samples, we use either human-annotated labels from the dataset or LLM-generated pseudo-labels to explore the effect of human annotation availability on our method.

Setup In distillation, we utilize gpt-3.5-turbo as the LLM[1]. We utilize 5-shot CoT prompting to enhance the quality and standardize the formats of the responses generated by the LLM. We follow the CoT prompts from Wei et al. (2022) for the CQA dataset and devise similar prompts for other datasets and self-evaluation. To strike a balance between diversity and cost, in the main experiment, we obtain five CoTs for each training instance and five self-evaluation outputs of each CoT from the gpt-3.5-turbo model and choose the T5-Base model (220M) (Raffel et al., 2020) as the SLM. We provide more experimental details in Appendix A. We also explore the effect of the value of the hyperparameter λ on the results, which are presented in Appendix B. Therefore, we select $\lambda = 0.5$ as the optimal hyperparameter for our main experiments. In all experiments, we report the mean results and standard deviations over 3 random runs.

4.1 Main results

Our results, presented in Table 1, show the advantages of our distillation method. Across all tasks and label types, the method we propose consistently outperformed the baselines (standard distillation and CoT distillation). In particular, we observe significant leaps in model performance when simultaneously training with five CoTs and their corresponding self-evaluation outputs. This reinforces our hypothesis about the value of incorporating self-evaluation and comprehensive thinking during the distillation process. Moreover, our approach

[1] Most experiments were conducted in August 2023 using the gpt-3.5-turbo model provided by the OpenAI API.

Method	SVAMP		CQA		ANLI	
	Pseudo-labels	Human-labels	Pseudo-labels	Human-labels	Pseudo-labels	Human-labels
Standard Distillation / Fine-tuning	49.2 ± 1.9	59.3 ± 1.2	58.7 ± 0.4	62.0 ± 0.4	37.7 ± 1.2	42.1 ± 5.0
1 CoT (i.e., CoT distillation)	51.7 ± 2.1	65.0 ± 1.1	59.7 ± 0.4	63.4 ± 0.2	39.8 ± 0.4	48.5 ± 1.2
1 CoT w/ Self-Evaluation	55.5 ± 0.4	67.8 ± 0.6	60.4 ± 0.2	63.7 ± 0.2	41.8 ± 0.4	49.2 ± 0.5
5 CoTs	54.8 ± 1.0	68.7 ± 0.2	61.2 ± 0.4	63.9 ± 0.2	41.7 ± 0.4	49.7 ± 0.8
5 CoTs w/ Self-Evaluation	**60.3 ± 0.6**	**72.7 ± 1.0**	**61.9 ± 0.3**	**65.0 ± 0.1**	**44.3 ± 0.2**	**50.8 ± 0.4**

Table 1: **Results of the main experiment.** We compare the accuracy (mean ± standard deviation, %) of different distillation methods on three different datasets (SVAMP, CQA, and ANLI) using 220M T5-Base models, utilizing pseudo-labels generated by the LLM or human-annotated labels. The Human-labels represent human-annotated labels. The "1 CoT" adopts the "Distilling step-by-step" method proposed by Hsieh et al. (2023).

exhibits a lower standard deviation than baseline methods, particularly under the "5 CoTs w/ self-evaluation" setting, indicating that our method offers stable improvements and enhances the robustness of distilled SLMs.

Effect of label quality A discernible pattern from the results is the gap in performance between models trained using LLM-generated pseudo-labels and human-annotated labels. Given the typically higher accuracy of human-annotated labels, which are considered the gold standard in supervised learning, this result is expected. However, regardless of the type of training labels used, our method exhibits consistent advantages, suggesting that the benefits of our distillation method are also robust to variations in label quality.

Robustness across tasks Our method's superiority is consistently evident when considering performance on different tasks, although the degree of improvement varies. In tasks such as MWPs (SVAMP dataset) and NLI (ANLI dataset), where reasoning complexity and potential for hallucinatory content are higher, the benefits of our methodology are more pronounced. This suggests that the proposed method effectively mitigates flawed reasoning and hallucinations in complex reasoning scenarios. In tasks like CQA (CQA dataset), where the reasoning processes might be less convoluted, the increments in performance are smaller yet still notable. This showcases the adaptability of our method to different types of reasoning complexity within various NLP tasks.

4.2 Effect of model size

To analyze the effectiveness of our proposed method across different model sizes, we further conducted experiments on the SVAMP dataset using both the T5-Small (60M) and T5-Large (770M) models. The results are presented in Table 2. Our method shows significant performance improvements on models of different sizes, reflecting the robustness of our method to model scale.

4.3 Effect of the number of CoTs

Using the SVAMP dataset as an example, we explore the effect of varying the number of CoTs on our method, where each CoT is accompanied by five self-evaluation outputs. As shown in Figure 3, initially, as the number of CoTs increases from 1 to 5, there is a notable improvement in performance metrics across both pseudo-labels and human-annotated labels. This trend underlines the benefit of exposing SLMs to a broader spectrum of reasoning processes and self-evaluation outputs, enhancing their capability to navigate complex reasoning and correct flawed reasoning. SVAMP as math word problems may benefit from a variety of different solutions, CQA as commonsense question answering may acquire richer knowledge from different answers, and ANLI as natural language inference might also benefit from different explanations. However, diminishing returns are observed when the number of CoTs exceeds five. In particular, when the number of CoTs exceeded 7, performance degradation is observed using human-annotated labels. It indicates that while multiple CoTs and self-evaluation outputs enrich the model's reasoning capabilities, there is a threshold beyond which performance cannot be further enhanced. This could be attributed to several factors: one possibility is that the integration of too many CoTs may introduce noise or conflicting reasoning patterns, thereby disrupting the distilled SLM. Another factor could be

Method	T5-Small		T5-Base		T5-Large	
	Pseudo-labels	Human-labels	Pseudo-labels	Human-labels	Pseudo-labels	Human-labels
Standard Distillation / Fine-tuning	25.5 ± 1.7	30.8 ± 1.6	49.2 ± 1.9	59.3 ± 1.2	60.2 ± 1.5	76.5 ± 1.2
1 CoT (i.e., CoT distillation)	29.2 ± 1.4	32.5 ± 0.4	51.7 ± 2.1	65.0 ± 1.1	66.2 ± 1.2	77.0 ± 1.2
1 CoT w/ Self-Evaluation	37.2 ± 1.4	35.2 ± 1.2	55.5 ± 0.4	67.8 ± 0.6	68.0 ± 1.1	79.0 ± 0.4
5 CoTs	36.5 ± 2.0	33.3 ± 1.0	54.8 ± 1.0	68.7 ± 0.2	66.5 ± 0.7	81.3 ± 0.8
5 CoTs w/ Self-Evaluation	**39.3 ± 1.2**	**36.8 ± 0.8**	**60.3 ± 0.6**	**72.7 ± 1.0**	**69.3 ± 0.6**	**83.7 ± 0.6**
Performance Gain	+ 10.1	+ 4.3	+ 8.6	+ 7.7	+ 3.1	+ 6.7

Table 2: **Experimental results for models of different sizes.** "Performance Gain" refers to the improvement in performance of our proposed method ("5 CoTs w/ Self-Evaluation") relative to the baseline method ("1 CoT").

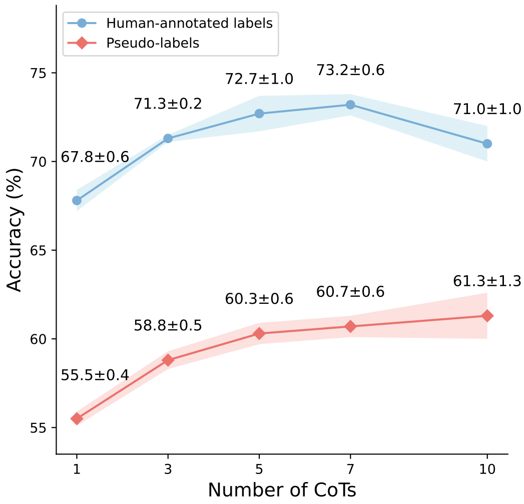

Figure 3: The experimental results of our method using the T5-Base model on the SVAMP dataset for different numbers of CoTs.

the cognitive load on the SLM. Beyond a certain scope, the model may struggle to effectively learn from additional training data.

This observation underscores the importance of finding an optimal balance in the number of CoTs used for distillation. As the number of CoTs and self-evaluation outputs increases, there is a corresponding rise in data costs and training expenses. Therefore, we opted to use five CoTs in our main experiments, balancing cost and performance.

4.4 Effect of the number of self-evaluation outputs for each CoT

We also investigate how the comprehensiveness of the self-evaluation affects the performance of our method by varying the number of self-evaluation outputs for each CoT. As shown in Figure 4, in both pseudo-labels and human-annotated labels settings, we can observe that: as the number of self-evaluation outputs per CoT increases, there is a notable enhancement in the accuracy of the distilled SLMs, although it may not strictly be monotonically increasing. This indicates that distilling more self-evaluation outputs enables SLMs to produce more accurate and reliable outputs. Notably, accuracy improves more with five CoTs than with a single CoT, underscoring the synergistic effect of combining multiple CoTs with corresponding self-evaluation. Overall, these findings emphasize the importance of incorporating self-evaluation in the distillation and confirm the value of introspec-

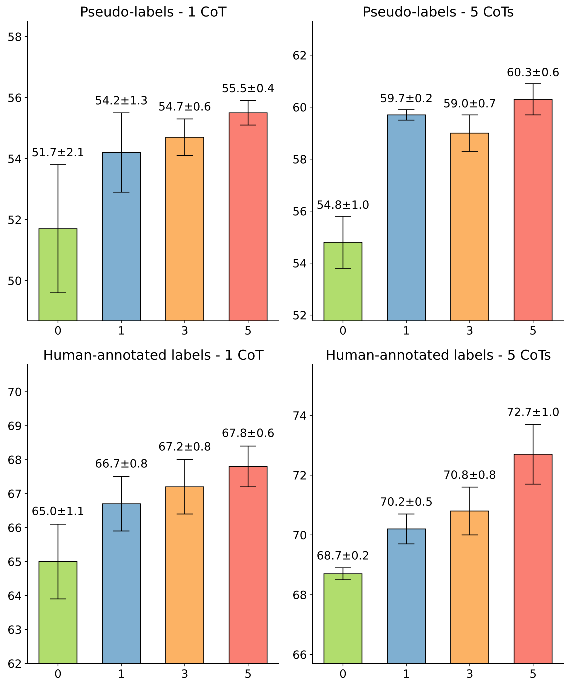

Figure 4: We present experimental results on the SVAMP dataset using the T5-Base model for different numbers of self-evaluation outputs for each CoT. Experiments are conducted under the settings of a single CoT and five CoTs, respectively.

ANLI Example	Ground Truth Label: neutral
Model Input	Premise: East Bengal Football Club is a professional football club, based in Kolkata, West Bengal, India. It currently competes in the I-League, the top tier of Indian football. The club has won three National Football League (India) (later named as I league) titles, eight Federation Cups, and three Indian Super Cups, among others. The club is the current holder of the Calcutta Football League. Hypothesis: All of the team members live in West Bengal.
Model Output (CoT Distillation)	Rationale: The premise mentions that East Bengal Football Club is based in Kolkata, West Bengal, India. The hypothesis states that all of the team members live in West Bengal. The hypothesis is consistent with the information provided in the premise. Label: entailment
Model Output (Our Method)	Rationale: The premise mentions that East Bengal Football Club is based in Kolkata, West Bengal, India. However, it does not provide any information about the location of the team members. Therefore, the premise neither supports nor contradicts the hypothesis. Label: neutral

Table 3: **A case study on the ANLI dataset.** The model trained by the CoT distillation method incorrectly predicts the label as entailment due to the premise location matching the hypothesis statement (highlighted in red), while the model trained by our method correctly identifies the lack of information regarding the team members' residences and correctly predicts the label as neutral (highlighted in blue).

tive self-evaluation in improving the reasoning and predictive capabilities of SLMs. Such introspective capabilities enable models to refine internal representations, rectifying possible misconceptions or potential pitfalls in their reasoning.

5 Discussion

5.1 Can our method mitigate the flawed reasoning and hallucinations of SLMs?

We conduct case studies on three datasets in the setting of using pseudo labels generated by LLMs. In the ANLI dataset case presented in Table 3, the task is to judge the relationship between the premise and hypothesis. The model trained by the CoT distillation method incorrectly infers that the premise entails the hypothesis because superficially the geographic locations mentioned in the two statements match each other. This flawed reasoning likely results from a lack of critical evaluation of the information's depth and relevance, a pitfall in models trained without a self-evaluation mechanism. Conversely, the model trained by our method identifies the lack of specific information about team members' residences in the premise and correctly concludes that the premise is neutral to the hypothesis. This accurate judgment showcases our method's strength in instilling a comprehensive and critical reasoning capability in the model, enabling it to discern the nuances and gaps in information that affect the reasoning. Case studies on other datasets are in Appendix C. The results indicate that our method effectively reduces flawed reasoning and hallucinations produced by distilled SLMs.

Given the absence of a gold standard for quantifying model hallucinations or harmful content, each of our 10 researchers (all holding Bachelor's degrees or higher) examined the outputs of different models for 200 pieces of data (with corresponding compensation). They manually compared the occurrences of hallucinations or harmful content in the outputs of models trained using our method and models trained using the CoT distillation baseline method. We statistically found that, on average, in approximately 7% of the cases, models trained with our method exhibited a significant reduction in hallucinations or harmful content, 91% of the cases tied and less than 2% contained more hallucinations or harmful content.

5.2 Can distilled SLMs really learn the self-evaluation capability?

Previous works (refer to Section 2) have already demonstrated that SLMs can achieve CoT reasoning by learning from the CoTs generated by teacher models. Based on this, we propose that SLMs should also be able to master a certain level of self-evaluation capability through learning from the self-evaluation outputs generated by teacher models. Gudibande et al. (2023) point out that "distilled imitation models are adept at mimicking ChatGPT's style but not its factuality", because crowd workers rate their outputs as competitive with ChatGPT, yet their performance on NLP benchmarks does not improve. However, our paper demonstrates through tests on three NLP benchmarks that our method significantly improves the performance of distilled SLMs. Therefore, the SLMs distilled by our method do not merely imitate the style of ChatGPT, but indeed enhance the model's capabilities. Furthermore, our study improves the capability of imitation models by using extensive imitation data in situations of limited resources and unchangeable base SLMs, which is consistent with the approach given by Gudibande et al. (2023) to improve the capability of imitation models.

In Appendix D, we tested SLMs trained with self-evaluation capability for their accuracy in evaluation predictions and printed their evaluation outputs. The results indicate that SLMs trained with self-evaluation capability achieve a consistency rate of approximately 90% with GPT-3.5 evaluations and are capable of producing rational evaluation processes. In contrast, SLMs without self-evaluation training were completely unable to perform evaluations.

Method	Reduced CQA	
	Pseudo-labels	Human-labels
Standard Distillation / Fine-tuning	41.6 ± 3.4	46.7 ± 1.2
1 CoT (i.e., CoT distillation)	45.1 ± 1.2	47.1 ± 1.5
1 CoT w/ Self-Evaluation	42.6 ± 2.0	45.9 ± 1.3
5 CoTs	44.8 ± 0.6	48.9 ± 1.6
5 CoTs w/ Self-Evaluation	**46.1 ± 0.1**	**49.0 ± 0.6**

Table 4: The experimental results of training using 900 samples from the CQA dataset.

5.3 What leads to differences in effectiveness?

Compared to CQA and ANLI, our method shows greater effectiveness on smaller SVAMP dataset. Is this due to the diminishing returns of our method as the volume of training data increases? We select the CQA dataset, which shows the least performance gain in our experiments, and reduce the number of training samples used from 8,766 to 900 to match the scale of SVAMP (keeping the test set unchanged) and then conduct experiments. The experimental results are presented in Table 4. Under the full training sample setting of the CQA dataset, "5 CoTs w/ Self-Evaluation" provides a performance gain of 2.2% and 1.6% respectively compared to "1 CoT" under two labels. In the setting of 900 training samples, the performance gains are 1.0% and 1.9% respectively. For the SVAMP dataset, the performance gains are 8.6% and 7.7%. Therefore, we believe that the returns of our method do not diminish with the increase in training data volume, but are more closely related to the nature of different tasks. SVAMP, as a math word problems task, is more likely to benefit from feedback through self-evaluation, while CQA, as a commonsense question answering task, benefits less. However, in our experiments, regardless of the task type, our method proved effective, demonstrating the universality of our approach.

Method	SVAMP	
	Pseudo-labels	Human-labels
5 CoTs	54.8 ± 1.0	68.7 ± 0.2
5 CoTs w/ Self-Evaluation	60.3 ± 0.6	**72.7 ± 1.0**
10 CoTs	55.8 ± 1.0	67.6 ± 0.2
10 CoTs w/ Self-Evaluation	**61.3 ± 1.3**	71.0 ± 1.0

Table 5: The experimental results of expanding the number of distilled CoTs to 10 CoTs on the SVAMP dataset.

5.4 Can learning self-evaluation be replaced by learning more CoTs?

From Table 5, it can be observed that the marginal gain of increasing from "5 CoTs" to "10 CoTs" is almost negligible, and the performance of "10 CoTs" is significantly lower than that of "5 CoTs w/ Self-Evaluation". In the case of "10 CoT", the incorporation of self-evaluation distillation still manages to enhance the performance of the model. Therefore, we further confirmed that the role of self-evaluation cannot be substituted by merely adding more CoT data. When increasing the number of CoTs is ineffective, employing our proposed method of distilling with self-evaluation can further enhance model performance, breaking through the performance ceiling of CoT distillation.

6 Conclusion

In this study, we have introduced an innovative method to effectively distill the more comprehensive capabilities from LLMs into SLMs, emphasizing both the transfer of self-evaluation capability and comprehensive thinking, to mitigate the shortcomings of previous CoT distillation methods. Comprehensive experiments demonstrate that our method outperforms prior distillation methods consistently in various NLP tasks, significantly improving the performance and reliability of SLMs. We hope that this study can promote the more effective and efficient utilization of SLMs, especially in resource-limited environments.

7 Limitations

Despite the promising results and advancements achieved in our study, certain limitations need acknowledgment and further investigation:

1. **Limited teacher and student models**: The experiments we conducted primarily utilized a single teacher model, GPT-3.5, and two student models, T5-Base and T5-Large. While these selections were influenced by their current popularity and efficacy, it is crucial to note that the landscape of LLMs and SLMs is rapidly evolving. As such, our distillation method may manifest differently when paired with other architectures or models. Future work will involve testing a wider range of models to confirm the universality of our method.

2. **Limited tasks**: Although we evaluated our methods on three different NLP tasks, NLP tasks are broad and complex. Therefore, future evaluations of our method's performance on a wider range of tasks are needed to provide a more comprehensive evaluation of its strengths and potential weaknesses.

3. **Self-evaluation reliability**: One inherent limitation of the self-evaluation process is its reliance on the LLM's capacity for introspection. If the LLM's self-evaluation mechanism is flawed or biased, it might adversely affect the distilled SLM. In future work, we will investigate the differences in self-evaluation capabilities among different LLMs, such as Llama 2 (Touvron et al., 2023), GPT-3.5, and GPT-4 (OpenAI, 2023), and how these differences affect the performance of distilled SLMs.

In conclusion, while we have made significant strides in advancing the distillation process from LLMs to SLMs, there exists a plethora of avenues for further refinement and exploration. Future endeavors should aim to address these limitations to ensure broader and more robust applicability.

8 Ethical Considerations

Potential risks While our approach is dedicated to reducing the flaws inherited by SLMs from LLMs, SLMs may still inherit harmful biases and discrimination from LLMs. Therefore, future work will aim to further minimize the impact of harmful content from LLMs on SLMs.

The use of closed source LLMs Many related studies and open source models have already utilized data obtained from the GPT family of models provided by OpenAI. We also obtain CoTs and self-evaluation outputs from the gpt-3.5-turbo model. However, the purpose of this study is not to develop models that compete with general large language models like ChatGPT. Instead, it aims to enhance the effectiveness and efficiency of small language models in resource-constrained environments, promoting the democratization of NLP. We only use gpt-3.5-turbo as the LLM to validate the effectiveness of our method, and it is not required to use the gpt-3.5-turbo model in practical applications, so different LLMs can be employed according to the licenses.

The use of AI assistants We employed ChatGPT to assist us in polishing our paper and writing code.

Acknowledgements

This research was partially supported by the Key R&D Program of Zhejiang under grant No. 2024SSYS0026.

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. Gkd: Generalized knowledge distillation for auto-regressive sequence models. *arXiv preprint arXiv:2306.13649*.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. 2023. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hongzhan Chen, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. 2023. MCC-KD: Multi-CoT consistent knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6805–6820, Singapore. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul

Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yao Fu, Hao-Chun Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

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. *arXiv preprint arXiv:2305.15717*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. 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, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv: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 *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pretrained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also “think” step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, 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)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023.

Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*.

David L Poole and Alan K Mackworth. 2010. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, Cambridge, UK.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach weaker agents? teacher explanations improve students via theory of mind. *arXiv preprint arXiv:2306.09299*.

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2269–2279, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023a. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. 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 *Proceedings of the 11th International Conference on Learning Representations, ICLR'2023*.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1948–1966, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xuekai Zhu, Biqing Qi, Kaiyan Zhang, Xingwei Long, and Bowen Zhou. 2023a. Pad: Program-aided distillation specializes large models in reasoning. *arXiv preprint arXiv:2305.13888*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023b. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

A Experimental details

Datasets The dataset statistics are shown in Table 6. Following Hsieh et al. (2023), for the SVAMP dataset, 20% of the original data is used as the test set. For the CQA dataset, the original validation set is used as the test set. Then, for both datasets, 10% of the data from the original training set is sampled to serve as the validation set. The ANLI dataset follows the original split. The language of all datasets is English. To the best of our knowledge, all datasets used have been widely employed in NLP research and do not contain any information that names or uniquely identifies individual people or offensive content.

Dataset	Train	Validation	Test
SVAMP	720	80	200
CQA	8,766	975	1,221
ANLI	16,946	1,000	1,000

Table 6: Dataset statistics.

LLM performance In Table 7, we report the accuracy of LLM (gpt-3.5-turbo) on three datasets in our experiments, including accuracy on the training set (i.e., the accuracy of pseudo-labels used for training SLMs) and accuracy on the test set.

Dataset	SVAMP	CQA	ANLI
Training Set	85.6	69.1	68.6
Test Set	84.3	72.4	55.1

Table 7: The accuracy (%) of LLM (gpt-3.5-turbo).

Models & Training The T5-Small[2] (60M), T5-Base[3] (220M) and T5-Large[4] (770M) models are all initialized with pre-trained weights obtained from Hugging Face, and the hyperparameter settings for their training are shown in Table 8. We perform the main experiments on 4 A100 GPUs.

B Effect of the hyperparameter λ

As shown in Figure 5, our experiments reveal trends regarding the effect of the hyperparameter λ on the accuracy of the SLMs trained using both pseudo-labels and human-annotated labels.

Hyperparameter	T5-Small / T5-Base	T5-Large
Total Batch Size	64	32
Learning Rate	5×10^{-5}	5×10^{-5}
Max Input Length	1,024	1,024
Maximum Steps (for SVAMP)	4,000	9,000
Maximum Steps (for CQA & ANLI)	12,000	-

Table 8: Training hyperparameter settings.

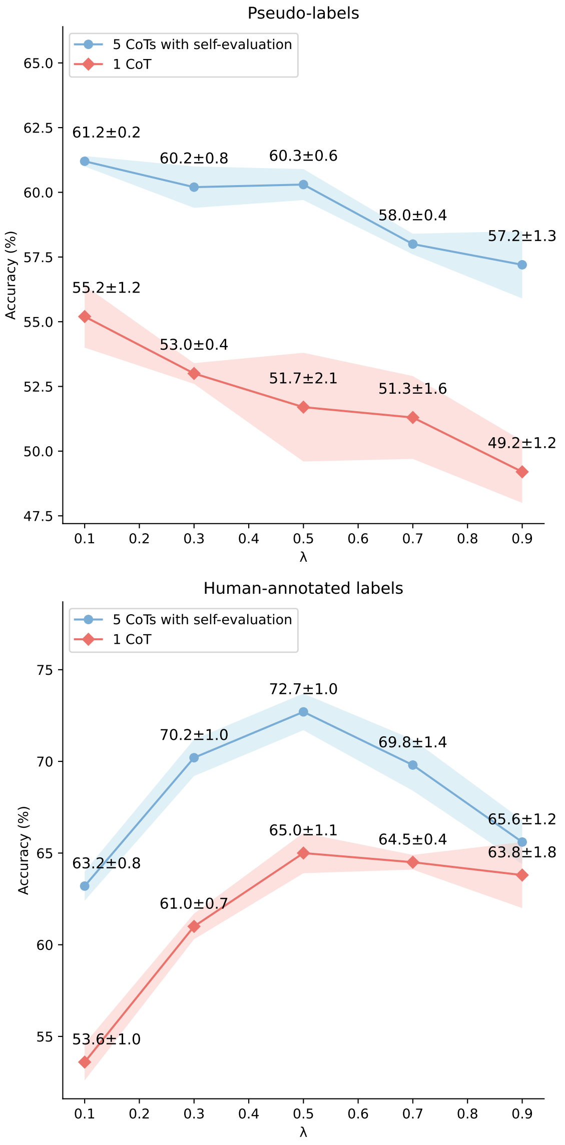

Figure 5: We present experimental results of distillation using the T5-Base model on the SVAMP dataset with different λ values for "1 CoT" and "5 CoTs with self-evaluation" respectively.

For pseudo-labels, the performance of both methods declines as λ increases, yet our approach exhibits a lesser decrease. Contrastingly, in the case

[2] https://huggingface.co/google/t5-v1_1-small

[3] https://huggingface.co/google/t5-v1_1-base

[4] https://huggingface.co/google/t5-v1_1-large

of human-annotated labels, we observe a different trend. The accuracy initially increases with λ, peaking at $\lambda = 0.5$, and then begins to decline. This pattern underscores a critical observation: up to a certain point ($\lambda \leq 0.5$), increasing the weight on human-annotated labels positively impacts the model's ability to predict labels. However, beyond this optimal point, overly emphasizing human-annotated labels while neglecting rationales can lead to a decrease in label prediction accuracy. This suggests that the best way to enhance model performance is to learn high-quality labels and rationales in a balanced way. The differing trends observed between pseudo-labels and human-annotated labels may be attributed to variations in label quality: human-annotated labels, being of higher quality, benefit the model's accuracy when their weight is increased, whereas low-quality pseudo labels do not require higher weighting.

Based on these observations, we select $\lambda = 0.5$ as the optimal hyperparameter for our main experiments, maintaining a balance between the weights of labels and rationales.

C Case study

The detailed case studies presented in Tables 3, 10, and 11 provide insightful examples demonstrating the effectiveness of our methodology compared to the baseline CoT distillation method. These cases highlight the importance of incorporating both self-evaluation and comprehensive thinking in the distillation process, which significantly reduces flawed reasoning and hallucinations in SLMs.

In the SVAMP example (Table 10), the model trained by the baseline CoT distillation method exhibits flawed reasoning in its calculation, erroneously summing the hours for learning Chinese and Spanish only, resulting in an incorrect total. This illustrates a common issue with CoT distillation, where the model may focus on a part of the problem, leading to incomplete reasoning. In stark contrast, the model trained by our method correctly identifies and sums the hours for all three languages, demonstrating a more comprehensive understanding and accurate reasoning process. This accurate reasoning underscores the effectiveness of our method, which incorporates both multiple CoTs and self-evaluation capability. By exposing the model to diverse reasoning processes and enabling it to evaluate its reasoning, our method equips the model to consider all relevant information comprehensively and to avoid flawed reasoning paths.

Similarly, in the CQA example (Table 11), the model trained by the baseline CoT distillation method incorrectly concludes that the most logical result of dying is a change of color, showcasing a case of flawed reasoning and hallucination. This error is likely due to a superficial association between the concepts of dying and color change, without a deeper understanding of the context of organic material decay. The model trained by our method, on the other hand, correctly identifies "death and decay" as the logical result of dying in the context of organic material, reflecting a deeper and more accurate comprehension of the question's essence. The comprehensive thinking instilled by our method, coupled with the ability to critically evaluate its reasoning, enables the model to select the most logical answer from the provided choices.

D Self-evaluation prediction accuracy and evaluation outputs of SLMs

Table 9 shows that the evaluation results of SLMs (T5-Base) are mostly consistent with those of GPT-3.5, indicating that they have learned most of the evaluation capabilities of GPT-3.5. Table 12 illustrates an example where the SLM trained with self-evaluation capability can conduct reasonable evaluation, whereas that without this training is unable to evaluate CoT. The outputs of SLMs in other samples are similar to the examples shown.

SVAMP	CQA	ANLI
84.7	94.1	91.8

Table 9: The self-evaluation prediction accuracy (%) of SLMs (T5-Base) trained with self-evaluation capability on three datasets.

SVAMP Example	Ground Truth Label: ((6.0 + 7.0) + 3.0)
Model Input	Every day Ryan spends 6 hours on learning English 7 hours on learning Chinese and 3 hours on learning Spanish. How many hours does he spend on learning English, Chinese, and Spanish in all?
Model Output (CoT Distillation)	Rationale: Ryan spends 6 hours on learning English and 7 hours on learning Chinese. To find out how many hours he spends on learning English, we need to add the number of hours spent on learning Chinese and Spanish together: 7 + 3 = 6. Label: (7 + 3)
Model Output (Our Method)	Rationale: Ryan spends 6 hours on learning English, 7 hours on learning Chinese, and 3 hours on learning Spanish in total. To find the total number of hours he spends on learning English, Chinese, and Spanish in all, we need to add the number of hours spent on learning English, Chinese, and Spanish together. So, the equation is 6 + 7 + 3. Label: (6 + 7) + 3

Table 10: **A case study on the SVAMP dataset.** The model trained by the CoT distillation method incorrectly calculates the total time to learn the three languages (highlighted in red), while the model trained by our method correctly sums the time to learn the three languages (highlighted in blue).

CQA Example	Ground Truth Label: death and decay
Model Input	Question: As a result of dying, what happens to organic material? Choices: [change of color, stop breathing, wake up, death and decay, getting cold]
Model Output (CoT Distillation)	Rationale: The question is asking about the result of dying. Of the given choices, the most logical result of dying is a change of color. Label: change of color
Model Output (Our Method)	Rationale: The question is asking about the result of dying and the process of organic material. Of the given choices, the most logical result of dying is death and decay, as organic material undergoes the process of decay. Label: death and decay

Table 11: **A case study on the CQA dataset.** The model trained by the CoT distillation method incorrectly considers that the most logical result of dying is a change of color (highlighted in red), while the model trained by our method correctly identifies the most logical result as death and decay (highlighted in blue).

SVAMP Self-Evaluation Example	Ground Truth Label: correct
Model Input	Paige and her mom planted 10 seeds in each flower bed, and there are 45 flower beds. So, they planted a total of 10 * 45 = 450 seeds. The answer is (10 * 45).
Model Output (w/o Self-Evaluation)	Rationale: . The answer is (10 * 45). So, they planted 450 seeds. So, they are 450 seeds. So, they are 450 seeds. So, they are 450 seeds. So, they are 450 seeds. So, they are. So. So. So. So. So. So Label: -
Model Output (w/ Self-Evaluation)	Rationale: Evaluation: The given answer correctly states that Paige and her mom planted 10 seeds in each flower bed and there are 45 flower beds. Then, it calculates the total number of seeds planted as 10 * 45 = 450 seeds, which is accurate. Finally, the answer represents these calculations in the form of an equation (10 * 45), which is correct. Label: correct

Table 12: **A case study on the evaluation of a CoT for the SVAMP dataset by SLMs trained with self-evaluation capability versus those not trained with self-evaluation capability.** The SLM trained with self-evaluation capability can output rationales and labels that are coherent and well-reasoned. In contrast, SLMs that are not trained with self-evaluation capability fail to produce logically sound rationales and formally appropriate labels.