Logic Consistency Makes Large Language Models Personalized Reasoning Teachers

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have advanced natural language processing significantly with Chain-of-Thought (CoT) reasoning and In-Context Learning (ICL), but their deployment is limited by high computational and operational costs. This paper introduces Personalized Chain-of-Thought Distillation (PeCoTD), a novel approach to transfer reasoning capabilities from LLMs to smaller, more deployable models. Recognizing the comprehension difficulties small LMs face with LLM-generated rationales, we first develop a metric called Self Logic Consistency (SLC) to assess rationale quality. This refinement process ensures the maintenance of semantic equivalence with the original LLM rationales, facilitating more effective fine-tuning and avoiding distribution shifts. This approach, focusing on data quality in Knowledge Distillation (KD), mitigates comprehension variability in small LMs and extends the applicability of CoT KD strategies. Our experiments show that PeCoTD significantly improves the reasoning abilities of small models across diverse datasets.

1 Introduction

Large language models (LLMs) have achieved state-of-the-art performances in many natural language processing tasks (Wang et al., 2019), due to emergence capabilities, such as Chain-of-Thought (CoT) (Wei et al., 2022a; Wang et al., 2023c; Kojima et al., 2022) capability and In-Context learning (ICL) (Brown et al., 2020; Min et al., 2022; Wang et al., 2023b) capability. To address complex tasks, LLM utilize their CoT capability, or reasoning capability to generate intermediate steps that lead to the final answer, referred to as the rationales. (Kojima et al., 2022) found that the CoT capability can be stimulated just by executing an instruction.

Nonetheless, a critical shortcoming of rationale-generating CoT reasoning approaches is their need for large models, with parameter counts reaching

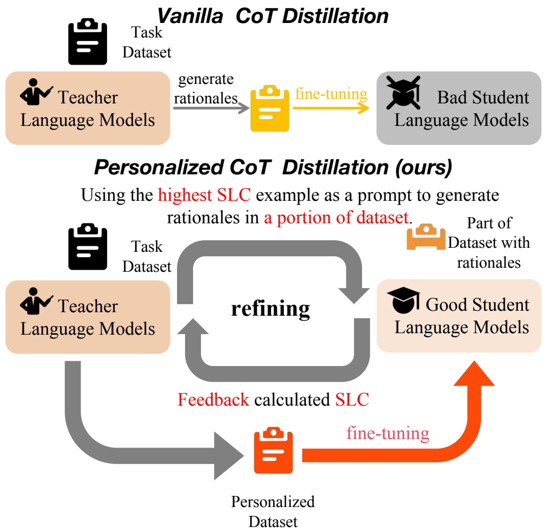

Figure 1: Personalized CoT Distillation Process. This illustrates our method, where high SLC scored examples are used as prompts to refine rationales, enhancing the training of student models through iterative refinements.

into the hundreds of billions (Wei et al., 2022b; Kojima et al., 2022). The scalable deployment of these models is significantly impeded by their formidable computational requisites and the substantial costs associated with inference.

Consequently, our endeavor is to facilitate same capable reasoning within small language models (small LMs), which present a more viable option for widespread deployment. Knowledge Distillation (KD) (Hinton et al., 2015) is a powerful tool to transfer the ability of large models (i.e., teacher models) into small ones (i.e., student models) with a minimal loss of reasoning capability.

The current KD of CoT capabilities are primarily based on rationale and black-box style. (Ho et al., 2023a) uses LLMs to generate rationales, then combines the rationales and answers together as completions for training small language models. (Hsieh et al., 2023) uses LLMs to generate rationales and answers separately, and then trains small language models step-by-step with the rationale and answer as the target objectives. SCoTD

(Li et al., 2023b) explored the factors influencing the KD of CoT capabilities through rationales, and concluded that the number of rationales is key to the distillation of CoT capabilities. However, it did not reveal the underlying reasons why the number of rationales is so significant.

Inspired by SCoTD (Li et al., 2023b), we implement 3 knowledge distillation strategies from GPT-3 to OPT-1.3B. The 'Random' strategy selects five random rationales from thirty, 'Diversity-a-b' uses SBERT for clustering and selects b rationales from a clusters, and 'All CoTs' uses all thirty rationales. Intuitively, because rationales contain more knowledge, KD with rationales helps small LM to better understand the world than KD with only label. Nonetheless, the experimental findings, as related to Figure 2, demonstrate that the performance of CoT distillation, when limited to merely five rationales, even though these cover the comprehensive knowledge contained within thirty rationales, significantly underperforms compared to CoT distillation using thirty rationales. When more rationales are selected uniformly within each cluster, the performance of CoT distillation approaches that of using all cot distillation. Therefore, for KD LLM into small LM, it is not enough for rationale to only include sufficient knowledge.

Notably, (Moschella et al., 2023) has observed that the representations within the latent space, generated across various training instances of neural networks, demonstrate significant variability. This indicates that the distribution of a small language model (small LM) significantly diverges from that of a large language model (LLM), needing an alignment to solve this problem (Yang et al., 2024). Consequently, this variability suggests that small LMs may not consistently interpret or "understand" the outputs from LLMs. Furthermore, the use of rationales generated by LLMs as training inputs for small LMs can lead to instances where these rationales are not effectively comprehended by the small LMs, resulting in limited training effectiveness.

In this paper, to address the challenge wherein small LMs struggle to comprehend the rationales produced by LLMs during CoT KD, we first modeled the relationship between the question, rationale, and label, identifying a metric called Self Logic Consistency (SLC) that effectively evaluates the quality of the rationale for student models. Secondly, we adopted a simple strategy, which

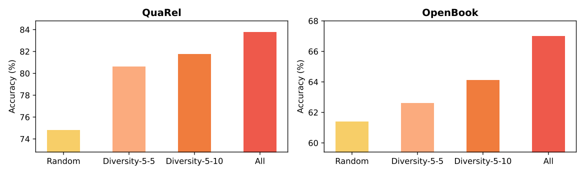

Figure 2: Comparison of CoT distillation using different rationale selection strategies.

we call *personalized Chain-of-Thought Distillation* (PeCoTD), requiring a personalized refining process to bridge the distribution gap between LLMs and small LMs.

We hypothesize that obstacles to rationale utilization stem from this distribution gap. To address the issue, as shown in Figure 1, PeCoTD uses a question-completion pair that aligns with the small model's learning distribution as a prompt to generate new rationales that maintain semantic equivalence with the original rationales offered by LLMs. This process, called refinement, can be repeated multiple times. After refinements, the refined rationales serve as surrogate targets during subsequent finetuning. Through this approach, PeCoTD inherently maintains the original distribution, avoiding distribution shifts and thereby better utilizing rationales. Given that PeCoTD focuses on data quality, it is orthogonal to the majority of CoT KD methods, making its applicability extensive.

Our experiments demonstrate that PeCoTD significantly improves the reasoning abilities of small models across various datasets. Specifically, PeCoTD demonstrates enhanced accuracy, improving T5-large by 1.75% on Strategy QA and 1.70% on CommonSense QA (CQA) as shown in Figure 3. Moreover, alignment with the teacher model is also improved, increasing similarity by 0.012 on Strategy QA and 0.011 on CQA as depicted in Figure 7, effectively bridging the distribution gap.

2 Method

2.1 Vanilla CoT Knowledge Distillation

The existing CoT (Chain of Thought) KD (Knowledge Distillation) generally utilizes both the CoT, referred to as rationale r_i generated by a teacher model $\mathcal{T}$ in response to a query q_i and the corresponding answer a_i. i is the index of the specific question. These elements are collectively used as the completion $c_i = (r_i, a_i)$ to fine-tune a small LM (Small Language Model), denoted as the student model $\mathcal{S}$ and parameterized by θ, as described in the equation below:

$$L_{\mathrm{FT}}(\theta) = -\sum_i \log \mathcal{S}_\theta(c_i \mid x_i), \qquad (1)$$

The method we follow to generate rationales and reformat them into prompt-completion pairs is based on the work of (Ho et al., 2023b). The final data prompt-completion pair, query q and completion c, takes the form of "<q_i> ###" and "<r_i> $--\,>$ <a_i> END".

2.2 Self-Logic Consistency (SLC)

In our work, we address the need for smaller language models (LMs) to be trained not merely on general rationales but on those that are most suited to their specific learning capacities. This approach is motivated by the observation that conventional training methods, which use a one-size-fits-all strategy for data, do not optimize for the internal cognitive structures of smaller models. To this end, we propose a refined metric based on Pointwise Mutual Information (PMI), which we term Self-Logic Consistency (SLC), to identify and utilize the most effective data for training these models.

Adapting PMI for Language Model Distillation. PMI is traditionally used in linguistic studies to measure the association between words within specific contexts. The standard PMI formula is given by:

$$\mathrm{PMI}(x, y) = \log \frac{P(x, y)}{P(x)P(y)} = \log \frac{P(y \mid x)}{P(y)} \quad (2)$$

We extend the application of PMI from simple word pairs to the evaluation of entire rationales. This adaptation is crucial as it allows us to gauge the coherence and relevance of the rationales generated by LLMs when used for training smaller models.

Self-Logic Consistency (SLC). Building directly on the PMI framework, the SLC score is a metric designed to specifically assess how well the rationales align with the intrinsic reasoning patterns of small LMs. It quantifies the suitability of each rationale by comparing the conditional and marginal probabilities of the completions:

$$\mathrm{SLC}_\theta(q_i, c_i) = \frac{P_\theta(c_i|q_i)}{P_\theta(c_i)} \quad (3)$$

Here, P_θ is the probability distribution estimated by the small model $\mathcal{S}_\theta$, q_i is a question, and c_i is its associated completion, including both a rationale r_i and an answer a_i. This metric leverages the internal logic of the small model to identify which data samples will best aid its learning process, reflecting a shift towards more personalized training approaches.

We employ the SLC metric to evaluate various generated rationales for each question. Based on these SLC scores, we curate datasets, separating the highest from the rest. This targeted selection ensures that smaller language models train on the most suitable rationales, potentially boosting their reasoning capabilities and effectiveness.

2.3 Iterative Refinement Using ICL

To further enhance the alignment between the rationales generated by the teacher model $\mathcal{T}$ and the student model $\mathcal{S}$'s learning capabilities, we implement an iterative refinement process using In-Context Learning (ICL). This approach leverages highest-scoring SLC samples as the prompts for generating new, more suitable rationales.

Initial Data Generation. We begin by generating a diverse set of rationales for the total dataset D_{total}. When generating rationales for the first time, we followed the zero-shot method of (Ho et al., 2023b). This initial generation is crucial for establishing a baseline of CoT distillation:

$$\{(q_i, r_{ij}, a_i)\}_{i=1}^{|\hat{D}_{\text{total}}|} \leftarrow \mathcal{T}(q_i)_{i=1}^{|\hat{D}_{\text{total}}|} \quad (4)$$

Here, $\hat{D}_{\text{total}}$ represents the total dataset with rationales; i indexes the question, and j indexes the rationales corresponding to each question, with a_i representing the answers.

Data Selection and Organization in Refinement After initial data generation, we compute the SLC scores for these rationales with the corresponding small LM.

The rationales of the highest scoring dataset $\hat{D}_{\text{h total}}$ is composed by selecting the rationale r with the highest SLC for each question q_i:

$$\hat{D}_{\text{h total}} = \{(q_i, r_i, a_i) \mid \begin{array}{l}\text{if } \mathrm{SLC}_\theta(q_i, c_i) \\ \text{is highest for } q_i \text{ in } \hat{D}_{\text{total}}\end{array}\} \quad (5)$$

Then five samples, question-completion pairs, are selected from $\hat{D}_{\text{h total}}$ as the context of prompt $\mathcal{P}_c$ for the next phase of rationales generation in refinement. Similarly, the rationales of $\hat{D}_{\text{l total}}$ is also composed of selecting the rationale r with the lowest SLC from each question q_i.

Using $\mathcal{P}_c$ as context, followed closely by a question q_i , through ICL, we employ the teacher model through ICL to generate new rationales that are more personalized to the student model's preferences:

$$\{(q_i, r_{ij}, a_i)\}_{i=1}^{|\hat{D}_{\text{total}}|} \leftarrow \mathcal{T}(\mathcal{P}_c, q_i)_{i=1}^{|\hat{D}_{\text{total}}|} \quad (6)$$

Each iteration aims to produce rationales that better aligns with the small model's intrinsic logic.

Method	Size	SLC	Single Eq	Add Sub	Multi Arith	GSM8K	SVAMP	Date Understanding	Last Letter	Coin Flip	Common SenseQA	Strategy QA
Random			0.00	0.00	0.00	0.00	0.00	17.12	0.00	50.00	20.00	50.00
Teacher: Llama2												
zero-shot			74.84	68.9	79.26	40.12	58.0	72.07	52.67	88.33	60.78	60.48
Student: GPT-2 (`small`, `medium`, `large`)												
(Ho et al., 2023b)	small-124M	low	1.32	3.36	7.22	2.12	8.67	19.82	5.33	56.0	24.0	48.33
		random	2.63	10.08	11.11	2.05	9.33	18.02	8	55.33	26.21	49.64
		high	3.29	5.04	8.89	3.03	8.33	17.12	7.33	57.33	27.6	50.95
	medium-354M	low	4.61	6.72	8.33	2.96	7.67	17.12	6.67	55.33	21.9	52.55
		random	2.63	5.04	11.11	2.65	8	15.32	7.33	54.67	23.36	49.34
		high	7.24	5.88	9.44	3.41	5.67	19.82	5.33	56.67	22.85	50.22
	large-774M	low	3.95	8.40	11.67	2.96	8	19.82	2.67	65.33	24.41	50.07
		random	1.97	5.88	7.78	3.18	6.67	14.41	4.67	66.67	25.5	51.82
		high	5.26	9.24	11.67	2.88	7.33	12.61	2.67	69.33	26.7	50.36
Student: T5 (`small`, `base`, `large`)												
(Ho et al., 2023b)	small-60.5M	low	1.97	3.36	6.11	2.20	7	19.82	12	82.33	35.14	50.36
		random	1.32	4.20	5.56	2.05	6.33	20.72	15.33	92	36.2	51.82
		high	4.61	4.20	5.0	2.35	5.33	26.13	15.33	92.67	37.1	53.57
	base-222	low	3.29	2.52	10.0	5.76	7.67	63.96	38.67	100	50.45	56
		random	3.29	3.36	12.22	6.37	9.33	68.47	40.67	100	51.68	56.62
		high	3.95	4.20	12.22	5.91	8.33	72.07	43.33	100	53.32	57.8
	large-737M	low	5.92	8.40	12.78	8.20	10.33	77.48	44.67	100	65.36	58.95
		random	7.90	10.92	11.11	8.41	11.67	81.08	44.67	100	66.75	59.24
		high	4.61	10.08	18.89	8.35	10.67	83.78	47.33	100	67.4	59.97

Table 1: Performance of T5 and GPT-2 model families across various datasets with different SLC levels. SLC levels are categorized as high, random, and low, indicating the highest, randomly selected, and lowest SLC scores for the CoTs generated by the teacher model for each question.

Iterative Process and Evaluation in Refinement. The iterative nature of this refinement allows for continual improvement. After generating new rationales, we reassess the SLC scores and select the highest scoring new samples for subsequent ICL. Notably, by focusing only on refinement, we can generate rationales for just 100 questions, thereby saving computational resources. The pseudocode for iterative refinement is shown in the Table 4.

3 Experiments

We empirically validate the effectiveness of our method. First, we demonstrated the impact of PMI on distillation performance. Second, we show that when compared to direct task distillation approaches, our method achieves better performance with much fewer number of training examples of diverse CoTs, and reduces the distribution gap between teacher and student models.

Setup. In the experiments, We use `Llama2-7B` (Touvron et al., 2023) as the teacher model to generate correct rationales or CoTs. We sample from `Llama2` with a temperature of $T = 0.9$. For each training example, we sample $N = 8$ rationales.

We use GPT-2 {Small, Medium, Large} (Radford et al., 2019) and T5 {Small, Base, Large} (Raffel et al., 2020) as representative model families for decoder-only and encoder-decoder architectures, respectively, serving as student models. All Student models is fine-tuned with a batch size of 32 and a learning rate of 6×10^{-5}.

Datasets. We evaluate our method on 10 datasets pertaining to four categories of complex reasoning, following (Kojima et al., 2022) These include arithmetic math word problems(SingleEq, AddSub, MultiArith, GSM8K, SVAMP(Patel et al., 2021)), other (Date Understanding), symbolic (Last Letter Concatenation, Coin Flip), and common sense (CommonSense QA(Talmor et al., 2019), Strategy QA) reasoning.

3.1 Results

Higher SLC Makes Better CoT Distillation Performance. As demonstrated in Table 1, higher SLC values correlate with enhanced CoT or reasoning capabilities in small LMs trained on datasets organized around selected rationales, except in cases where performance approach randomness. For example, in Strategy QA and CQA datasets, the accuracy differences between the lowest and highest SLC levels for the T5-large model are 1.02% and 2.04%, respectively. However, across various arithmetic math word problem datasets (SingleEq, AddSub, MultiArith, GSM8K, SVAMP), small LMs consistently exhibit poor performance,

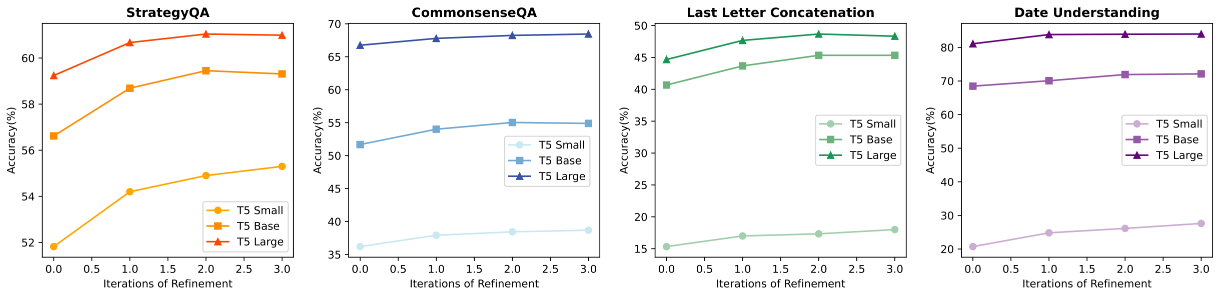

Figure 3: The performance of PeCoTD across the small, base, large size of T5 on datasets such as Date Understanding, Last Letter Concatenation, CQA, and Strategy QA improves as the number of refinement iterations increases.

regardless of the dataset size. This indicates that a relatively small number of parameters is inadequate for effectively handling arithmetic math word problems.

Regrettably, GPT-2 {Small, Medium, Large} performs almost at random across all datasets. Nonetheless, the decoder-only model definitely possess CoT capability. We think the poor performance of GPT-2 is primarily due to the insufficient number of parameters. Further experiments and analysis is in Sec 4.

The Encoder-Decoder Style Have More Reasoning Capability in Small Size. However, we observed that the GPT-2 family's performance across all datasets approximated random responses. This indicates that GPT-2 lacks CoT capabilities in small size. Meanwhile, we observed that within the T5 family, there is a notable enhancement in CoT capabilities with an increase in parameter count in Date Understanding, Last Letter Concatenation, Coin Flip, CQA, and Strategy QA. This demonstrates that SLC is effective in assessing the quality of CoT data for small models. This indicates that, when sizes are equal and small, the encoder-decoder models demonstrates greater reasoning capabilities than the decoder-only models.

Considering the structure of SLC, when the conditional probability $P_\theta(c_i \mid q_i)$ exceeds the marginal probability $P_\theta(c_i)$, it implies that the incorporation of the question q_i positively influences the likelihood of generating the completion c_i. This relationship indicates a higher logical consistency between the question and the completion, thereby suggesting a stronger SLC. Conversely, if $P_\theta(c_i \mid q_i)$ is less than $P_\theta(c_i)$, it implies that, particularly for smaller language models, there is a logical discrepancy between the question and its completion. This inconsistency results in a reduced SLC.

Figure 4 compares SLC scores across four datasets: CQA, Strategy QA, Last Letter, and Date Understanding, using T5 large and GPT-2 large. The T5 model consistently scores above 1 across all datasets, with scores in the Date Understanding dataset surpassing 2.0. This indicates strong and coherent logical consistency in T5's responses, particularly evident in its robust handling of diverse prompts. This suggests that the encoder-decoder architecture is better suited to adapting across diverse distributions.

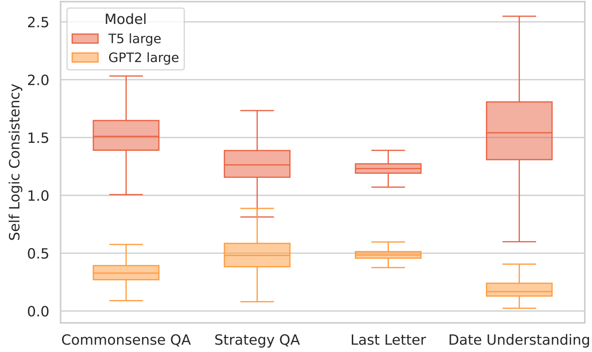

Figure 4: Comparison of SLC scores between T5-large and GPT2-large on the datasets of CQA, Strategy QA, Last Letter Concatenation, and Date Understanding.

In contrast, the GPT-2 model scores below 1.0 in all datasets, suggesting its struggle to align reasoning effectively with the posed questions. The consistently low scores and minimal variability point to a significant limitation in handling tasks that demand nuanced understanding and logical reasoning, likely due to its decoder-only architecture.

PeCoTD Help Small LM Reason. Since our method focuses on data, while other existing CoT methods concentrate on fine-tuning approaches, our method is orthogonal to other CoT KD methods. Therefore, our method is compared solely with the baseline (Ho et al., 2023b) to demonstrate its effectiveness in data operations.

Considering that small LMs struggle with arithmetic math word problems but attain 100% accuracy on Coin Flip, we exclusively present PeCoTD's performance on datasets: Strategy QA, CQA, Last Letter Concatenation, and Date Under-

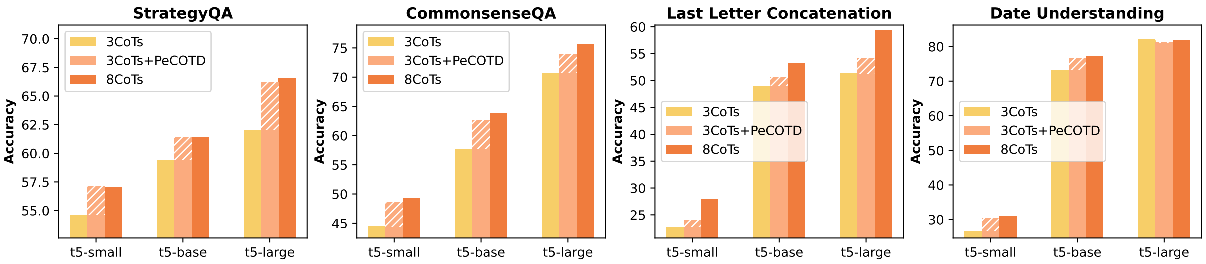

Figure 5: The performance of the T5 {Small, Base, Large} on the datasets for Date Understanding, Last Letter Concatenation, CQA, and Strategy QA, CoT KD with 3 CoTs, PeCoTD method with 3 CoTs, and CoT KD with eight CoTs.

standing. For simplicity and clarity, we will only display the performance of T5, which is shown in Figure 3. When the number of iterations is 0, we randomly select from the rational generated by the teacher model.

Figure 3 shows that as the number of refinement iterations increases, the performance of CoT distillation improves. For instance, following 3 refinement iterations, the T5-large model demonstrated accuracy improvements of 1.75%, 1.70%, 3.66%, and 2.90% across four respective datasets. This indicates that the teacher model has generated rationales that are more suitable for small LMs to learn from, aligning better with the distribution of small models.

Larger Language Models Possess a Greater Capability to Bridge the Distribution Gap. Figure 3 reveals that as the number of parameters increases, the marginal benefit from iterative refinement decreases. This indicates that models with larger parameter quantities are more adept at learning information from texts with varying distributions. One possible reason is that larger models, characterized by increased complexity and parameter count, exhibit enhanced capability in capturing subtle nuances and intricate patterns present within disparate datasets. This heightened capacity facilitates the seamless transition of knowledge from one distribution to another. (Goyal et al., 2024) also pointed out that in models with a large number of parameters, the model will converge to the same level regardless of the quality of the training data, as long as sufficient training resources are available, which confirms our point of view.

Fewer CoTs with PeCoTD is Near Equivalent to More CoTs. Following (Li et al., 2023b), we use SBERT (Reimers and Gurevych, 2019) to calculate embeddings for the rationales and apply hierarchical clustering to organize the eight rationales per question into 3 clusters, selecting one rationale from each. These selected rationales are used for

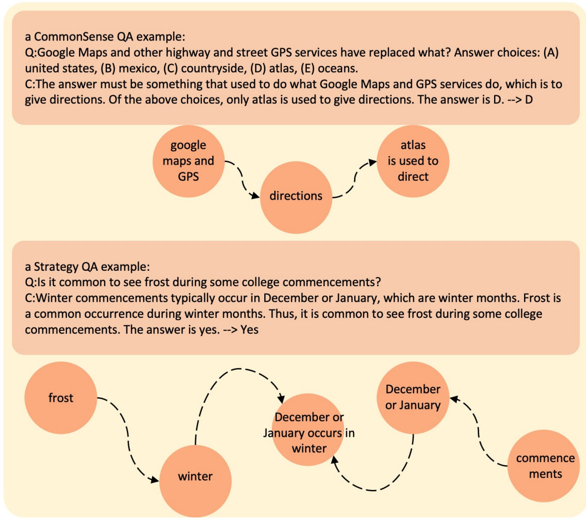

Figure 6: Comparison of a typical CQA example with a typical Strategy QA example. Every example contains a question and a completion.

vanilla CoT KD. Similarly, we use the PeCoTD method to produce rationales, employing the rationales derived from each cluster of each question for CoT KD. We set the PeCoTD refinement iterations to 3. In the final experimental group, CoT KD is conducted using all eight rationales. All results are illustrated in Figure 5. We find that in most cases, PeCoTD has led to improvements.

When analyzing the Strategy QA results, the performance of PeCoTD with 3 CoTs closely approaches that of vanilla CoT KD with eight CoTs. However, in the case of CQA, while the PeCoTD method with 3 CoTs shows improved performance over the vanilla CoT KD, it still lags behind the vanilla CoT KD with eight CoTs by gaps of 0.59%, 1.19%, and 1.66% for the small, base, and large sizes, respectively.

Upon examining the two datasets, we identified distinct differences. CQA is knowledge-intensive, relying more on whether the model possesses relevant knowledge. In contrast, Strategy QA is primarily reasoning-intensive, demanding robust logical capabilities of the model.

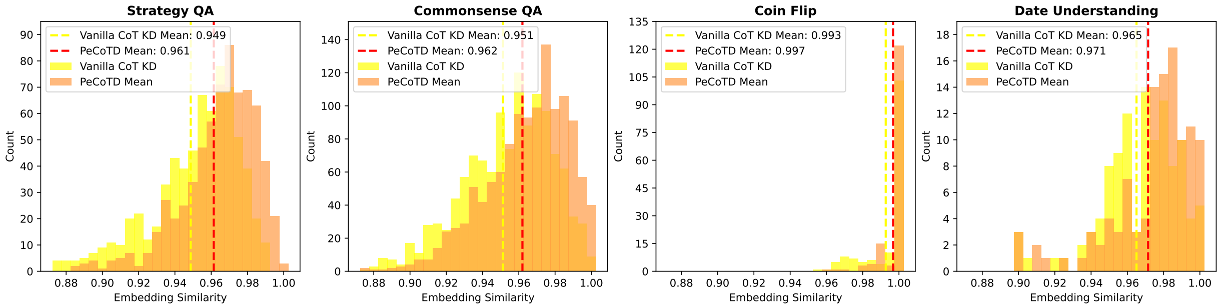

Figure 7: Distribution of cosine similarity between SBERT embeddings of completions from teacher and student models across various tasks. The tasks include CQA, Strategy QA, Date Understanding, and Coin Flip. PeCoTD shows higher overall similarity, indicating better alignment between teacher and student models.

In Figure 6, we display a representative example from CQA alongside one from Strategy QA. (Wang et al., 2024) revealed that there actually exists a directed graph in the corpus consisting of concepts as points and relationships as edges, where the chain-of-thoughts is paths in this directed graph. Hence, we also constructed the corresponding reasoning paths based on the completions. From Figure 6, we can see that the inference in the CQA example is simpler, with fewer edges representing reasoning; Conversely, the Strategy QA example has more edges, suggesting a greater presence of reasoning, commonly associated with multi-hop questions. This suggests that questions in Strategy QA lead to more complex rationales generated by the teacher model, resulting in a distribution of rationales that is too broad for the student model to effectively learn. However, PeCoTD is specifically designed to bridge distribution gaps. The larger the gap, the more significant the enhancements PeCoTD can provide. This explains why PeCoTD achieves greater enhancements in Strategy QA compared to CQA.

4 Analysis

Decoder-Only Can Also Logic-Consist. Undoubtedly, decoder-only style language models possess chain-of-thought capabilities. We did not observe clear chain-of-thought performance on GPT-2, possibly due to scaling law. The size of {Small, Medium, Large} is not yet sufficient to possess chain-of-thought ability. However, we find that math word problems are too challenging for small language models to perform effectively. Therefore, we employ GPT-2 XL, with 1.6 billion parameters, to conduct experiments on the other five datasets, as shown in Table 2.

SLC	Date Understanding	Last Letter	Coin Flip	Common SenseQA	Strategy QA
	17.12	0.00	50.00	20.00	50.00
Student: GPT-2 XL					
low	13.51	5.0	72.0	25.31	49.05
random	11.71	6.33	75.33	28.42	49.34
high	15.32	4.0	78.0	29.48	49.2

Table 2: The performance of GPT-2 xl with different levels in SLC on Date Understanding, Last Letter, Coin Flip, CQA, and Strategy QA.

As illustrated in Table 2, while GPT-2 XL achieves non-zero scores in Date Understanding, Strategy QA and Last Letter Concatenation, the results are not substantial enough to conclusively demonstrate problem-solving capabilities, suggesting that the model may have recognized only superficial patterns. In contrast, in the Coin Flip task, GPT-2 XL, getting score of 75.33% with randomly selected data, appears to have effectively learned to predict the final state of the coin, likely due to the simplicity of the problem. For CQA, GPT-2 XL's performance significantly deviates from randomness, showing a clear improvement between the lowest and highest SLC with 4.17%.

Iterations of Refinement	Date Understanding	Last Letter	Coin Flip	Common SenseQA	Strategy QA
0	11.71	6.33	75.33	28.42	49.34
1	15.32	3.67	87.67 (+12.34)	34.66 (+6.24)	49.64
2	8.11	7.33	96.0 (+20.67)	38.19 (+9.77)	52.69
3	12.61	8.67	100 (+24.67)	35.98 (+7.56)	50.51

Table 3: Performance of GPT-2 XL across various datasets with increasing refinement iterations.

As shown in Table. 3, due to GPT-2 XL's failure to effectively solve the Date Understanding and Last Letter Concatenation tasks, refining the rationales used to train student models does not enhance the model's performance. However, in the Coin Flip and Strategy QA tasks, although each refinement iteration enhances model performance, the marginal gains diminish with each subsequent iteration. For CQA, the third refinement failed to benefit the student model and even resulted in a 4.3% decrease in performance.

Feedback Helps LLM learn the Distribution

of the Small LM. On the test dataset, we conduct inferences using four models: the vanilla teacher model, the teacher model with refined prompts, the student model of T5-base trained with vanilla CoT KD, and the student model of T5-base trained with personalized CoT via PeCoTD. The number of iterations of refinement is 3. Subsequently, we calculate the cosine similarity between the embeddings from the SBERT outputs of the completions from each teacher-student pair. The distribution of these similarities is shown in Figure 7. For clarity and simplicity, we only present data from CQA, Strategy QA, Date Understanding, and Coin Flip.

Intuitively, a higher cosine similarity between the teacher and student models' output indicates a narrower distribution gap. In the case of CQA, Strategy QA, and Date Understanding, PeCoTD results in the overall similarity shifting towards 1.0, with the mean increasing by 0.012, 0.011, and 0.05. Despite the simplicity of the Coin Flip task and the small distribution gap between teacher and student models, PeCoTD still manages to enhance similarity.

SLC Affects Length of Rationales Differently in Different Architecture. When employing small LMs to evaluate the rationales generated by teacher models, it has been observed that for models within the T5 family, rationales selected with higher SLC are notably shorter. Conversely, in the case of the GPT-2 family, higher SLC correlates with longer rationales. It's worth noting that they may not be the longest or shortest among all rationales. This should be related to the decoder-only style and the encoder-decoder style.

For instance, for Strategy QA, the average lengths and SLCs of rationales selected by different models are shown in the Figure 8. We observed that for both T5 and GPT-2 models, the average length of rationales filtered by each SLC level is converging. Moreover, further statistic analysis revealed that their overlap is very high in the same style models. For instance, in the Strategy QA, the GPT models small, medium, large simultaneously select 62.63% of the data when the SLC is low, and 60.37% of the data when the SLC is high, which is shown in Figure 9. This suggests that models of the same style exhibit consistency in their rationales selection.

5 Related Work

CoT represents intermediate reasoning steps from problem to answer, encompassing logical relationships and knowledge concepts. (Li et al., 2022) enhances smaller reasoning models by leveraging explanations from large language models (LLMs) in a multi-task learning approach, boosting their reasoning and explanatory capabilities. (Magister et al., 2023) explores transferring these reasoning skills to smaller models via knowledge distillation, balancing model size and dataset for optimal reasoning skills. (Fu et al., 2023) advocates fine-tuning instruction-tuned models, distilling CoT reasoning trajectories from larger models to enhance task performance outside the training distribution. (Geva et al., 2022) incorporates LLM-generated rationales into a multi-task training regimen for smaller models. (Ho et al., 2023a; Li et al., 2023b) use explicit CoTs for CoT distillation. (Shridhar et al., 2023) develops two specialized models: a problem decomposer and a subproblem solver. The decomposer breaks down problems into subproblems, while the solver focuses on these segments. (Wang et al., 2023a) uses contrastive decoding to ensure rationales are relevant to their corresponding answers, promoting appropriate and counterfactual reasoning. (Zhu et al., 2023) enables learner models to benefit from program-aided reasoning, detecting and correcting erroneous reasoning steps.

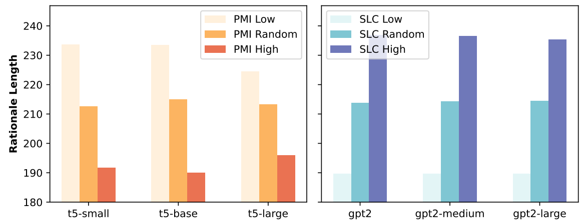

Figure 8: Comparison of the average length of CoTs about Strategy QA for T5 and GPT-2 under different levels of SLC.

6 Conclusion

In this study, we introduced a data-focused methodology called Personalized Chain-of-Thought Distillation that enhances the reasoning capabilities of small LMs by personalizing rationales from LLMs based on self logic consistency. Our results demonstrate significant improvements in reasoning accuracy across multiple datasets compared to existing methods. PeCoTD effectively reduces the distribution gap between teacher and student models, ensuring that small LMs not only receive information but also effectively understand and utilize it. We look forward to further explorations into the scalability of this approach and its broader application potential.

7 Limitations

Generalization for All Size of Models and Types of Datasets. While our study introduces Personalized Chain-of-Thought Distillation (PeCoTD) as an effective method, several limitations related to its generalization across different model sizes and datasets warrant attention. Firstly, the effectiveness of PeCoTD heavily depends on the initial quality and diversity of rationales generated by the teacher model. his dependency may limit its utility when the teacher model's outputs are suboptimal or lack sufficient diversity, particularly affecting its performance across various model sizes. Secondly, although PeCoTD enhances small language models' performance, it does not uniformly address all types of reasoning tasks. This is evident from the varied performance observed across different datasets, indicating that tasks requiring advanced mathematical reasoning or extensive factual knowledge pose significant challenges that PeCoTD may not fully overcome. Additionally, the approach's scalability and effectiveness across different model sizes and task complexities remain constrained, highlighting the need for further research to improve its generalization capabilities.

The Computational Demands. Resource for computing, though reduced compared to training large models directly, remain significant, especially when scaling up to larger datasets or more complex model architectures. This scalability issue could limit practical deployments in resource-constrained environments. Lastly, the refinement process within PeCoTD, though effective, introduces additional complexity in training workflows, which might complicate its adoption without specialized knowledge or adjustments in existing infrastructure.

Does SLC Accurately Reflect Suitability for Small LMs? Although higher SLC showed higher performance in CoT distillation, we cannot intuitively let small LMs tell us that they prefer rationales with higher SLC. Socreval (He et al., 2023) employed ChatGPT to evaluate the quality of rationales from multiple dimensions. We adopted a similar approach, assessing rationales based on our criteria: logical coherence, comprehensibility, and the use of advanced vocabulary. Our results indicate that the scoring outcomes are largely random, regardless of the rationales' effectiveness in aiding small model learning. Furthermore, the scores generated by ChatGPT reflect its biases rather than the suitability of small models, as these models often fail to provide consistent evaluations based on the given prompts. Consequently, it is valuable to explore methods that enable small models to express in human language which rationale they prefer.

References

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.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *CoRR*, abs/2301.12726.

Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. 2022. Lm-debugger: An interactive tool for inspection and intervention in transformer-based language models. *arXiv preprint arXiv:2204.12130*.

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. 2024. Scaling laws for data filtering–data curation cannot be compute agnostic. *arXiv preprint arXiv:2404.07177*.

Hangfeng He, Hongming Zhang, and Dan Roth. 2023. Socreval: Large language models with the socratic method for reference-free reasoning evaluation. *arXiv preprint arXiv:2310.00074*.

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. 2023a. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14852–14882. Association for Computational Linguistics.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023b. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14852–14882. Association for Computational Linguistics.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. 2023a. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. *Advances in Neural Information Processing Systems*, 36:72888–72903.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023b. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2665–2679. Association for Computational Linguistics.

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.

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.

Sewon Min, Xinxi Lyu, et al. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *NAACL-HLT*, pages 2080–2094. Association for Computational Linguistics.

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. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

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, Toronto, Canada, July 9-14, 2023*, pages 7059–7073. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT (1)*, pages 4149–4158. 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*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5546–5558. Association for Computational Linguistics.

Xinyi Wang, Alfonso Amayuelas, Kexun Zhang, Liangming Pan, Wenhu Chen, and William Yang Wang. 2024. Understanding the reasoning ability of language models from the perspective of reasoning paths aggregation. *arXiv preprint arXiv:2402.03268*.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023b. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *CoRR*, abs/2301.11916.

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

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

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

Zhaorui Yang, Qian Liu, Tianyu Pang, Han Wang, Haozhe Feng, Minfeng Zhu, and Wei Chen. 2024. Self-distillation bridges distribution gap in language model fine-tuning. *arXiv preprint arXiv:2402.13669*.

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

A Appendix

Iterative Refinement Process
Initialize parameters: $iterations$, D_{total}, $\mathcal{T}$ `(teacher model)`, $\mathcal{S}$ `(student model)`
Step 1: Initial Data Generation Generate initial rationales for D_{total}: $\hat{D}_{\text{total}} \leftarrow \mathcal{T}(D_{\text{total}})$

Step 2: Iterative Refinement
for $i = 1$ **to** $iterations$ **do**
Compute SLC scores: $SLC_\theta(q_i, c_i)$
Select highest SLC rationales:
$\hat{D}_{\text{h total}} = \{(q_i, r_i, a_i) \mid$ if $SLC_\theta(q_i, c_i)$ is highest for q_i in $\hat{D}_{\text{total}}\}$
Select five samples as prompt context P_c from $\hat{D}_{\text{h total}}$
Generate new rationales using $\mathcal{T}$ with P_c:
$\hat{D}_{\text{total}} \leftarrow \mathcal{T}(P_c, q_i)$
end for

Step 3: Final Model Training Compute final SLC scores: $SLC_\theta(q_i, c_i)$ Select best rationales for training: $\hat{D}_{\text{final}} = \{(q_i, r_i, a_i) \mid SLC_\theta(q_i, c_i)$ is highest$\}$ Fine-tune student model $\mathcal{S}$ with $\hat{D}_{\text{final}}$

Table 4: Pseudocode for Iterative Refinement Process

Repetition of teacher model's output results in degeneration. When there are many repetitions in the rationales generated by the teacher model, the student model will generate more repetitions. (Li et al., 2023a) also found the similar phenomenon.

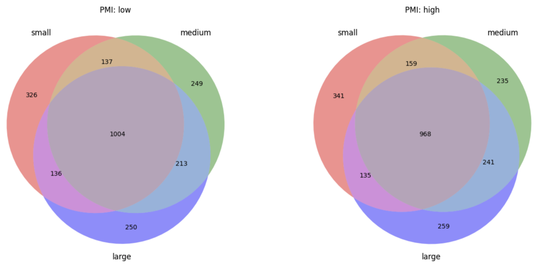

Figure 9: When SLC is low and high, the overlap of rationales selected by GPT-2 small, medium, and large. The count of each part represents the number of rationales for that section.

For example, when the training data is highly repetitive and has an average length of 473, after training, the T5-small model outputs an average length of 1419. By reading the outputs, we found that in a large proportion, the trained T5-small continuously generates repetitive sentences until it exceeds the `max_length` parameter, thereby avoiding the generation of answers.