Unveiling the Key Factors for Distilling Chain-of-Thought Reasoning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel in reasoning tasks through Chain-of-Thought (CoT) prompting. However, CoT prompting greatly increases computational demands, which has prompted growing interest in distilling CoT capabilities into Small Language Models (SLMs). This study systematically examines the factors influencing CoT distillation, including the choice of granularity, format and teacher model. Through experiments involving four teacher models and seven student models across seven mathematical and commonsense reasoning datasets, we uncover three key findings: (1) Unlike LLMs, SLMs exhibit a *non-monotonic* relationship with granularity, with stronger models benefiting from finer-grained reasoning and weaker models performing better with simpler CoT supervision; (2) CoT format significantly impacts LLMs but has *minimal* effect on SLMs, likely due to their reliance on supervised fine-tuning rather than pretraining preferences; (3) Stronger teacher models do *NOT* always produce better student models, as diversity and complexity in CoT supervision can outweigh accuracy alone. These findings emphasize the need to tailor CoT strategies to specific student model, offering actionable insights for optimizing CoT distillation in SLMs.

1 Introduction

Large Language Models (LLMs) have demonstrated exceptional capabilities through extensive pretraining on diverse human language data (Brown et al., 2020; Hoffmann et al., 2022; Team et al., 2024a; Meta, 2024a; OpenAI, 2024). Chain-of-Thought (CoT) prompting has further enhanced their abilities by guiding LLMs to generate intermediate reasoning tokens, which emulate human cognitive processes and improve interpretability (Kojima et al., 2022; Wei et al., 2023; Lyu et al., 2023). Advances in CoT prompting have explored techniques like extending reasoning steps (Jin et al., 2024; Merrill and Sabharwal, 2024) and refining reasoning formats (Deng et al., 2024; Xu et al., 2024a). However, CoT's token-intensive nature significantly increases computational demands (Zhao et al., 2024), limiting its practicality in resource-constrained settings. This has spurred interest in distilling CoT capabilities into Small Language Models (SLMs) as a more efficient alternative (Team et al., 2024b; Meta, 2024b).

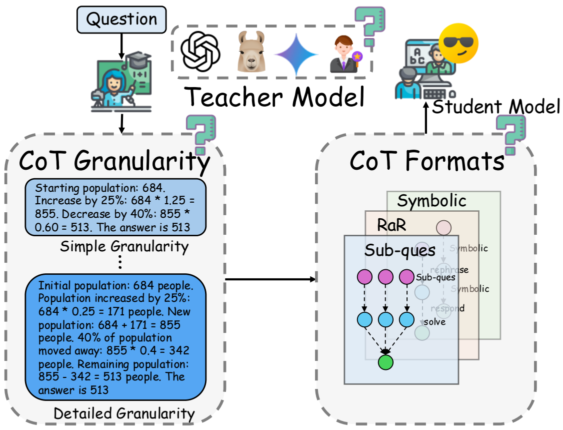


Figure 1: Overview of CoT Distillation. Different teacher models generate CoT supervision with varying levels of granularity and formats to fine-tune the student model.

Since SLMs often struggle to independently generate effective CoT reasoning solutions (Kaplan et al., 2020; Stolfo et al., 2023), distilling CoT capabilities requires fine-tuning SLMs on teacher-annotated CoT datasets, where the teacher can be either human experts or more powerful LLMs. While previous research has demonstrated successful distillation of CoT capabilities into SLMs (Ho et al., 2023; Magister et al., 2023; Xu et al., 2024b; DeepSeek-AI et al., 2025), the choice of teacher annotators and CoT generation methods has often been arbitrary. A critical, yet unexplored, research question remains: *What is the most effective CoT supervision for training a student model to develop robust reasoning capabilities?* Analogous to how human teachers instruct students, there are three key factors that influence how effectively a student absorbs knowledge:

- **Choice of teacher**: This defines *who* teaches the student. Different teachers bring varying levels of knowledge, teaching styles, and problem-solving approaches. In reality a student's performance can vary significantly depending on the teacher, and the most knowledgeable person is not always the best teacher.
- **Granularity of teaching**: This defines *what* level of detail is provided. Teachers may provide varying levels of explanation: some offer detailed, step-by-step reasoning, while others skip over simpler steps, assuming they are self-evident. The optimal level of granularity depends on the student's perspective of what needs to be explained.
- **Format of teaching**: This defines *how* the reasoning is structured and presented. Even with the same teacher and granularity level, the way information is organized and expressed can significantly impact learning outcomes. Some students may prefer plain language explanations, while others may thrive with more technical, mathematical language.

Building on this analogy of how human teaching impacts student performance, we conducted extensive experiments on four mathematical reasoning datasets of varying difficulty and three commonsense reasoning datasets, using four teacher models to distill reasoning skills to seven student models. We adopted a 1-shot prompting approach for generating CoT annotations, which we found to be the most effective in maintaining consistency in teaching style while controlling granularity. Our key findings are: (1) While LLMs benefit monotonically from detailed steps, SLMs exhibit an *non-monotonic relationship*. Stronger student models benefit from finer granularity, while weaker ones can be overwhelmed by excessive explanations and prefer simpler CoT annotations; (2) CoT format changes influence LLMs, likely due to their pretraining preferences for certain structures, but this effect is *less pronounced* in SLMs, which adapt more readily to diverse formats during fine-tuning; (3) Contrary to prior research suggesting that better teacher models invariably lead to better student performance (Zong et al., 2023), in the task of distilling CoT capabilities, we find that *better teacher models do not always produce better student models*. Sronger student models benefit more from advanced teacher model. Human-annotated CoTs, despite their near-perfect accuracy, often underperform LLM-generated CoTs. Our work presents the first systematic framework for optimizing CoT distillation, laying the groundwork for enhancing the reasoning capabilities of SLMs.

2 Related Work

CoT prompting CoT prompting (Wei et al., 2023) has become a pivotal technique for enhancing reasoning capabilities in LLMs by introducing intermediate reasoning steps. Automated approaches like Auto-CoT (Zhang et al., 2023), Tree-of-Thoughts (Yao et al., 2023) and Self-play Mutual Reasoning (Qi et al., 2024) explore multiple reasoning paths to expand the search space and improve task accuracy. These methods focus on increasing the reasoning length or expanding the reasoning horizon to handle complex tasks. Recent studies have underscored the importance of reasoning granularity and formats in enhancing LLM performance. For instance, Jin et al. (2024) identified that longer reasoning steps improve task success for complex problems, while overly concise steps can reduce effectiveness. Tailored reasoning formats(Khot et al., 2023; Zhou et al., 2023; Deng et al., 2024; Xu et al., 2024a) have demonstrated substantial improvements across tasks. However, these reasoning optimization strategies often comes with significant computational costs (Nayab et al., 2024), raising concerns about the trade-off between accuracy and efficiency.

Knowledge distillation While direct prompting enables LLMs to perform complex reasoning through CoT, SLMs struggle due to limited capacity (Stolfo et al., 2023). Knowledge distillation (KD) provides an effective framework for transferring the reasoning capabilities of teachers to SLMs (Xu et al., 2024b). A simple yet effective approach is using a teacher-student paradigm, which employs teacher-generated CoT steps to guide SLMs, addressing their limitations and enhancing reasoning-intensive task performance (Magister et al., 2023; Ho et al., 2023; Shridhar et al., 2023). Despite these advances, a systematic exploration of how to balance reasoning granularity, format, and teaching strategies remains lacking. Addressing these gaps is crucial for optimizing CoT distillation and enabling efficient reasoning in SLMs.

3 Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_1^N$ denote a reasoning dataset with N (x_i, y_i) pairs. Chain-of-Thought distillation aims to train a student S to generate intermediate reasoning steps C_i for each input x_i in order to generate the right y_i. The optimal C_i to train the student model is influenced by three key pedagogical factors: **Choice of Teacher**, **Granularity of Teaching**, and **Format of Teaching**:

Choice of Teacher The teacher model T generates a reasoning chain $\mathcal{C}_T(x_i)$ for each input x_i, which guides the student model in producing the correct answer. The teacher can either be an LLM or a human with varying styles and expertise.

Granularity of Teaching Granularity refers to the level of detail in the CoT reasoning. A high-granularity annotation provides detailed, step-by-step reasoning, while a low-granularity annotation skips steps and provides a more abstract summary. We represent the CoT chain with granularity g as $\mathcal{C}_g(x_i) = (c_{g,1}, c_{g,2}, \ldots, c_{g,k_g})$ where k_g is the number of reasoning steps. Higher k_g and more tokens in $c_{g,i}$ indicates higher granularity.

Format of Teaching Format refers to the structure in which the CoT reasoning is presented. It could be in natural language, formal logic, or symbolic representation. We denote the CoT chain in format f as $\mathcal{C}_f(x_i)$. The format impacts how the reasoning steps are conveyed.

Given these three factors, the distillation process involves supervised fine-tuning of the student model S on generated CoT annotations:

$$\mathcal{L}_{\text{distill}} = \sum_{i=1}^{N} \mathcal{L}(S(x_i), \mathcal{C}_{T,g,f}(x_i) \oplus y_i)$$

where $S(x_i)$ is the generation from S, $\mathcal{C}_{T,g,f}(x_i)$ denotes the CoT annotation generated under teacher T with granularity g and format f, $\oplus$ denotes concatenation and $\mathcal{L}$ measures the discrepancy between $S(x_i)$ and the ground truth.

4 Experimental setup

4.1 Generation of CoT Annotation

Teacher Models We use three teacher models: **GPT-4o**(OpenAI, 2024), **Gemini-1.5-Flash**(Team et al., 2024a), and **LLaMA 3 70B**(Meta, 2024a), chosen for their diverse architectures and reasoning capacities. Additionally, we include **human-annotated** CoTs, typically considered the ground-truth reasoning steps (Kumar et al., 2024).

Generation Method The CoT generation process begins with selecting a representative problem from the training split as a 1-shot example. This example is used to prompt teacher models for generating annotations under various configurations. For granularity, we prompt the model to generate CoTs with varying levels of detail simultaneously. For format, we prompt the model to generate CoTs for each format individually. [1] Details regarding the workflow, prompt designs, and case studies are included respectively in Appendix C and D.

4.2 Tasks and Datasets

Mathematical Reasoning To evaluate mathematical reasoning, we utilize four datasets with varying complexity levels: **SVAMP** (Patel et al., 2021), **GSM8K** (Cobbe et al., 2021), **AQuA-RAT** (Ling et al., 2017), **MATH** (Hendrycks et al., 2021). SVAMP, GSM8K, and MATH require numerical answers, while AQuA-RAT adopts a multiple-choice format. From the MATH dataset, we randomly sample problems from subcategories such as prealgebra, algebra, number theory, and counting and probability, ensuring a representative coverage of diverse mathematical domains.

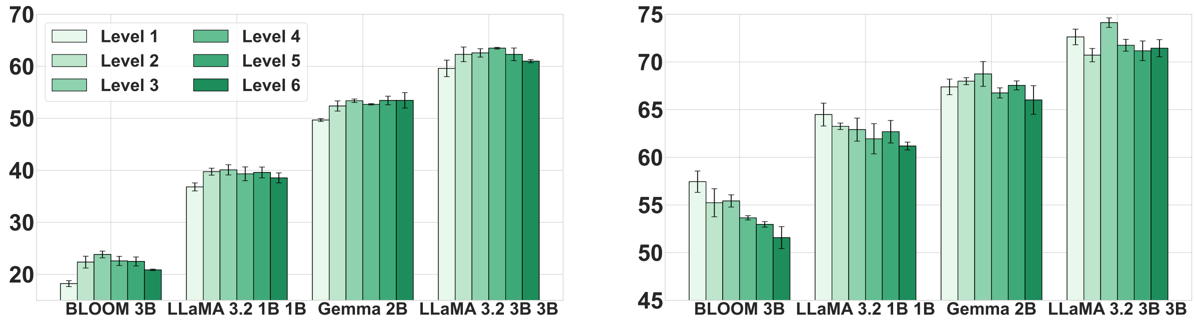
Commonsense Reasoning For commonsense reasoning, we use three datasets: **CommonsenseQA** (CSQA, Talmor et al. 2019; Aggarwal et al. 2021), **OpenBookQA** (OBQA, Mihaylov et al. 2018), and **StrategyQA** (STQA, Geva et al. 2021). These datasets test the models' ability to handle everyday reasoning and general knowledge tasks. CSQA uses a 5-class multiple-choice format, OBQA has 4 classes, and STQA is binary.

For evaluation, answers are extracted from generated responses using predefined templates and regular expressions. We use accuracy as our evaluation metric, which is calculated as the ratio of correctly predicted instances to the total number of instances: Accuracy $= N_{\text{correct}}/N_{\text{total}}$. The complete details can be found in the Appendix A.

5 Effects of Granularity

While previous research has shown that increasing reasoning granularity improves LLM performance

[1] For granularity, we also investigated other data collection strategies, such as generating reasoning steps forward and backward simultaneously, but these methods did not produce better data, as shown in Appendix C.1.



(a) GSM8K

(b) CommonsenseQA

Figure 2: Performance of student models with different granularity. Most models achieve peak accuracy at intermediate granularity levels.

through detailed intermediate steps (Jin et al., 2024; Merrill and Sabharwal, 2024), SLMs differ fundamentally from LLMs in their ability to process complex reasoning chains. This raises a critical question: *does increasing reasoning granularity still yield consistent benefits for SLMs in the task of CoT distillation?* In this section, we investigate this question using GPT-4o as the teacher model.

Non-Monotonic Scaling in Student Models As shown in Figure 1, our experiments reveal a *non-monotonic relationship between CoT granularity and student model accuracy*. Most models exhibit peak performance at intermediate granularity levels. Further increasing granularity leads to diminishing returns and even performance declines. It suggests that intermediate granularity strikes a balance between informativeness and efficiency in CoT, whereas overly detailed reasoning chains may introduce redundant information which is overwhelming especially for weaker models.

Table 1 presents the performance of three representative student models across seven evaluation datasets. We include a baseline called *Only Answer*, where student models are fine-tuned to predict answers without CoT. Similar to System 1's automatic thinking (Yu et al., 2024), higher baseline score suggests that the model may have implicitly learned the relevant knowledge during pretraining (Prabhakar et al., 2024).

Notably, *stronger and more recent student models*, such as those from the Gemma and LLaMA family, achieve significant performance gains from KD at *higher granularity levels*. In contrast, *smaller and weaker models* like BLOOM family improve on simpler tasks such *at the intermediate granularity levels* but struggle on more challenging datasets, sometimes performing no better than random guessing. This trend of BLOOM family aligns with parameter scaling laws (Kaplan et al., 2020) for simpler tasks but breaks down for more complex ones, where smaller models fail to acquire the reasoning abilities due to limited training data. Full results are provided in Appendix E.

These findings emphasize that CoT granularity plays a crucial role in CoT distillation. Customizing granularity levels to align with the student's abilities is thus critical for maximizing the efficiency and effectiveness.

Distinguishing Granularity from Length Effect Increasing reasoning granularity often leads to longer sequences as a byproduct. To isolate the impact of granularity from sequence length, we pad CoT training samples for a lower granularity level g_1 with non-informative filler content to match the sequence length of a higher granularity g_2, such that $\text{avg\_len}(\mathcal{D}_{g_1}) \approx \text{avg\_len}(\mathcal{D}_{g_2})$. This modification allows us to assess whether reasoning accuracy stems from granularity or simply sequence length. The specific padding procedure can be found in the Appendix F.

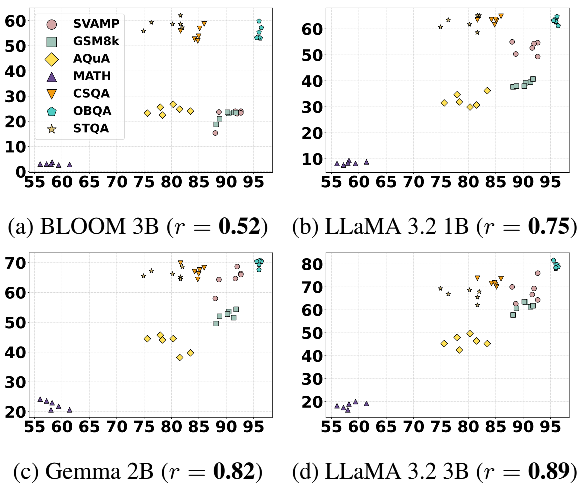
As shown in Table 2, padding Level 1 reasoning chains to level 5 consistently failed to replicate the gains observed with actual higher-granularity reasoning, which demonstrates that *simply increasing sequence length without introducing meaningful reasoning steps does not enhance model performance*. Furthermore, adding filler content may introduce noise or distract the model, leading to degraded performance (Zhou et al., 2024; Li et al., 2024). This highlights the critical role of granularity, rather than sequence length alone, in driving reasoning efficacy.

Dataset	Only Answer	Gemma 2B Performance					
		Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
SVAMP	47.70	$59_{\pm 4.58}$	$64.33_{\pm 0.00}$	$65.22_{\pm 0.69}$	$65.89_{\pm 0.38}$	$\mathbf{67.11}_{\pm \mathbf{1.35}}$ $^{\uparrow 13.74\%}$	$66.89_{\pm 1.02}$
GSM8K	8.20	$49.66_{\pm 0.27}$	$52.36_{\pm 0.98}$	$53.37_{\pm 0.33}$	$52.69_{\pm 0.13}$	$53.42_{\pm 0.83}$	$\mathbf{53.45}_{\pm \mathbf{1.48}}$ $^{\uparrow 7.63\%}$
AQuA-RAT	20.47	$40.68_{\pm 1.27}$	$42.91_{\pm 1.42}$	$43.7_{\pm 2.58}$	$39.9_{\pm 1.49}$	$\mathbf{44.88}_{\pm 0.79}$ $^{\uparrow 12.48\%}$	$44.49_{\pm 2.36}$
MATH	9.00	$23.4_{\pm 1.06}$	$21.53_{\pm 2.16}$	$\mathbf{24.4}_{\pm \mathbf{0.20}}$ $^{\uparrow 16.19\%}$	$21.93_{\pm 0.42}$	$23.0_{\pm 1.22}$	$21.0_{\pm 0.69}$
CSQA	**69.86**	$67.38_{\pm 0.82}$	$67.98_{\pm 0.37}$	$68.74_{\pm 1.30}$	$66.75_{\pm 0.53}$	$67.54_{\pm 0.47}$	$66.01_{\pm 1.50}$
OBQA	69.60	$71.53_{\pm 1.94}$	$69.93_{\pm 0.90}$	$69.93_{\pm 1.36}$	$68.33_{\pm 1.27}$	$\mathbf{72.00}_{\pm \mathbf{1.64}}$ $^{\uparrow 5.37\%}$	$70.13_{\pm 1.62}$
STQA	60.69	$\mathbf{67.59}_{\pm \mathbf{1.04}}$ $^{\uparrow 7.11\%}$	$63.1_{\pm 1.79}$	$64.6_{\pm 1.56}$	$63.45_{\pm 1.24}$	$65.75_{\pm 1.77}$	$64.14_{\pm 1.58}$
Dataset	**Only Answer**	**LLaMA 3.2 3B Performance**					
		Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
SVAMP	53.70	$69_{\pm 3.61}$	$65.89_{\pm 3.53}$	$68.33_{\pm 1.86}$	$68.11_{\pm 3.27}$	$69.78_{\pm 1.07}$	$\mathbf{74.33}_{\pm \mathbf{1.45}}$ $^{\uparrow 12.81\%}$
GSM8K	9.30	$59.59_{\pm 1.59}$	$62.29_{\pm 1.41}$	$62.57_{\pm 0.80}$	$\mathbf{63.48}_{\pm \mathbf{0.16}}$ $^{\uparrow 6.53\%}$	$62.29_{\pm 1.22}$	$60.98_{\pm 0.31}$
AQuA-RAT	19.60	$44.36_{\pm 2.31}$	$44.88_{\pm 2.19}$	$45.01_{\pm 2.37}$	$46.19_{\pm 3.94}$	$\mathbf{47.24}_{\pm \mathbf{4.77}}$ $^{\uparrow 6.49\%}$	$46.33_{\pm 3.01}$
MATH	9.40	$19.07_{\pm 0.90}$	$19.6_{\pm 1.06}$	$19.73_{\pm 1.72}$	$\mathbf{20.27}_{\pm \mathbf{1.42}}$ $^{\uparrow 11.37\%}$	$19.93_{\pm 2.20}$	$18.2_{\pm 1.64}$
CSQA	62.00	$72.62_{\pm 0.82}$	$70.71_{\pm 0.70}$	$\mathbf{74.12}_{\pm \mathbf{0.50}}$ $^{\uparrow 4.82\%}$	$71.75_{\pm 0.62}$	$71.17_{\pm 1.03}$	$71.44_{\pm 0.90}$
OBQA	74.40	$79.33_{\pm 0.42}$	$79.73_{\pm 0.70}$	$78.8_{\pm 0.80}$	$77.8_{\pm 0.92}$	$79.27_{\pm 2.04}$	$\mathbf{80.2}_{\pm \mathbf{1.78}}$ $^{\uparrow 3.08\%}$
STQA	55.52	$66.44_{\pm 1.55}$	$62.76_{\pm 2.82}$	$67.47_{\pm 1.44}$	$66.78_{\pm 1.39}$	$63.91_{\pm 2.79}$	$\mathbf{68.62}_{\pm \mathbf{1.20}}$ $^{\uparrow 9.34\%}$
Dataset	**Only Answer**	**BLOOM 3B Performance**					
		Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
SVAMP	5.00	$15.44_{\pm 0.51}$	$23.67_{\pm 0.00}$	$23.11_{\pm 1.26}$	$\mathbf{24.00}_{\pm \mathbf{0.67}}$ $^{\uparrow 55.44\%}$	$22.22_{\pm 0.69}$	$22.22_{\pm 1.02}$
GSM8K	4.60	$18.2_{\pm 0.57}$	$22.34_{\pm 1.14}$	$\mathbf{23.81}_{\pm \mathbf{0.65}}$ $^{\uparrow 30.82\%}$	$22.57_{\pm 0.88}$	$22.47_{\pm 0.86}$	$20.85_{\pm 0.15}$
AQuA-RAT	**28.00**	$24.67_{\pm 0.82}$	$24.41_{\pm 1.72}$	$20.34_{\pm 1.82}$	$26.90_{\pm 2.41}$	$25.85_{\pm 0.45}$	$24.28_{\pm 2.17}$
MATH	**4.60**	$3.2_{\pm 1.04}$	$2.8_{\pm 0.40}$	$2.33_{\pm 0.61}$	$2.73_{\pm 0.23}$	$3.53_{\pm 0.50}$	$2.8_{\pm 0.20}$
CSQA	20.56	$\mathbf{57.44}_{\pm \mathbf{1.12}}$ $^{\uparrow 11.38\%}$	$55.23_{\pm 1.47}$	$55.42_{\pm 0.64}$	$53.65_{\pm 0.22}$	$52.96_{\pm 0.29}$	$51.57_{\pm 1.15}$
OBQA	37.80	$57.2_{\pm 1.59}$	$52.33_{\pm 0.61}$	$54.87_{\pm 2.02}$	$54.6_{\pm 1.04}$	$\mathbf{57.47}_{\pm \mathbf{2.64}}$ $^{\uparrow 9.82\%}$	$52.93_{\pm 1.81}$
STQA	54.14	$58.85_{\pm 1.74}$	$\mathbf{61.04}_{\pm \mathbf{3.06}}$ $^{\uparrow 3.72\%}$	$60.58_{\pm 3.09}$	$59.89_{\pm 3.13}$	$59.19_{\pm 1.21}$	$59.08_{\pm 2.87}$

Table 1: Performance of Gemma 2B, LLaMA 3.2 3B and BLOOM 3B at six granularity levels. For each dataset, the best performance is boldfaced, and red text shows the relative improvement (%) for highest vs. lowest performance in six levels. *Only Answer*: Student models are fine-tuned to directly predict answers without CoT.

Granularity	GSM8K		AQuA-RAT	
	Acc	Seq. Length	Acc	Seq. Length
Level 1	47.61	100.93	40.15	149.31
Level 1 Padded	46.62	143.43	37.80	220.34
Level 5	**52.92**	138.16	**42.51**	216.13

Table 2: Performance and sequence length of Gemma 2B on GSM8k and AQuA-RAT with varying granularity levels and padding conditions.



(a) BLOOM 3B ($r = \mathbf{0.52}$) (b) LLaMA 3.2 1B ($r = \mathbf{0.75}$)

(c) Gemma 2B ($r = \mathbf{0.82}$) (d) LLaMA 3.2 3B ($r = \mathbf{0.89}$)

Figure 3: Scatter plots of teacher (GPT-4o, x-axis) vs. student accuracy (y-axis) across datasets and granularity levels. Each point marker represents a specific dataset.

Correlation between Granularity and Student Models Figure 3 illustrates the overall relationship between teacher and student performance across datasets at varying granularity levels. We further list the Pearson correlation coefficient (r) for each student model. The results reveal a clear trend: *as student model capacity increases, its performance aligns more closely with the teacher model's preferences for reasoning granularity*

Stronger student models demonstrate significantly higher alignment with the teacher's optimal granularity, indicating better transferability of reasoning structures. In contrast, weaker student models show a lower correlation, suggesting the limited ability to adapt to the teacher's granularity preferences. This highlights the importance of tailoring granularity configurations to *match the capabilities of student models*, rather than relying solely on the teacher's performance trends.

Conclusion

Increasing CoT granularity does not lead to monotonic improvements in student models. Stronger models benefit from higher granularity, whereas weaker models peak at intermediate levels and struggle with complex tasks. Optimizing granularity based on student capacity, rather than uniformly following the teacher model, is key to maximizing CoT distillation efficiency.

Dataset	CoT Format	BLOOM 560M	BLOOM 1.1B	BLOOM 1.7B	BLOOM 3B	Gemma 2B	LLaMA 3.2 1B	LLaMA 3.2 3B
SVAMP	Original CoT	$5.56_{\pm 2.41}$	$\mathbf{10.67}_{\pm 1.00}$	$\mathbf{16.56}_{\pm 0.51}$	$22.22_{\pm 0.69}$	$\mathbf{67.11}_{\pm 1.35}$	$52.44_{\pm 1.71}$	$69.78_{\pm 1.07}$
	Least-to-most	$\mathbf{6.11}_{\pm 1.07}\uparrow$	$10.44_{\pm 0.69}\downarrow$	$14.67_{\pm 1.00}\downarrow$	$24.00_{\pm 1.45}\uparrow$	$66.56_{\pm 0.69}\downarrow$	$54.44_{\pm 1.26}\uparrow$	$\mathbf{75.00}_{\pm 0.67}\uparrow$
	RaR	$4.89_{\pm 0.19}\downarrow$	$9.00_{\pm 0.58}\downarrow$	$14.11_{\pm 0.69}\downarrow$	$\mathbf{24.22}_{\pm 1.58}\uparrow$	$65.67_{\pm 1.73}\downarrow$	$\mathbf{54.56}_{\pm 0.69}\uparrow$	$73.89_{\pm 1.26}\uparrow$
	Symbolic CoT	$5.89_{\pm 0.51}\uparrow$	$6.44_{\pm 0.38}\downarrow$	$9.00_{\pm 0.67}\downarrow$	$19.22_{\pm 1.07}\downarrow$	$64.78_{\pm 0.77}\downarrow$	$51.78_{\pm 1.07}\downarrow$	$72.89_{\pm 1.50}\uparrow$
GSM8K	Original CoT	$\mathbf{8.19}_{\pm 0.27}$	$13.09_{\pm 0.83}$	$\mathbf{16.86}_{\pm 1.25}$	$\mathbf{22.47}_{\pm 0.86}$	$\mathbf{53.42}_{\pm 0.83}$	$\mathbf{39.58}_{\pm 1.04}$	$62.29_{\pm 1.22}$
	Least-to-most	$7.88_{\pm 0.35}\downarrow$	$\mathbf{13.52}_{\pm 0.87}\uparrow$	$15.54_{\pm 0.72}\downarrow$	$21.86_{\pm 0.56}\downarrow$	$51.93_{\pm 0.07}\downarrow$	$39.25_{\pm 1.10}\downarrow$	$62.07_{\pm 0.70}\downarrow$
	RaR	$5.89_{\pm 0.22}\downarrow$	$10.84_{\pm 0.40}\downarrow$	$13.72_{\pm 0.59}\downarrow$	$20.02_{\pm 0.27}\downarrow$	$51.99_{\pm 1.22}\downarrow$	$38.09_{\pm 0.46}\downarrow$	$\mathbf{63.02}_{\pm 0.56}\uparrow$
	Symbolic CoT	$5.94_{\pm 0.62}\downarrow$	$10.74_{\pm 0.64}\downarrow$	$13.27_{\pm 0.20}\downarrow$	$19.33_{\pm 0.73}\downarrow$	$47.12_{\pm 0.39}\downarrow$	$34.70_{\pm 0.89}\downarrow$	$58.94_{\pm 0.83}\downarrow$
AQuA	Original CoT	$18.64_{\pm 1.98}$	$21.92_{\pm 3.66}$	$22.31_{\pm 1.38}$	$\mathbf{25.85}_{\pm 0.45}$	$\mathbf{44.88}_{\pm 0.79}$	$\mathbf{33.20}_{\pm 2.17}$	$\mathbf{47.24}_{\pm 4.77}$
	Least-to-most	$19.69_{\pm 3.22}\uparrow$	$20.73_{\pm 0.91}\downarrow$	$23.10_{\pm 2.17}\uparrow$	$24.41_{\pm 1.42}\downarrow$	$38.32_{\pm 1.86}\downarrow$	$28.48_{\pm 2.17}\downarrow$	$41.60_{\pm 2.77}\downarrow$
	RaR	$\mathbf{21.26}_{\pm 3.94}\uparrow$	$\mathbf{22.57}_{\pm 2.79}\uparrow$	$\mathbf{24.28}_{\pm 2.50}\uparrow$	$25.07_{\pm 3.94}\downarrow$	$41.86_{\pm 3.16}\downarrow$	$31.10_{\pm 2.39}\downarrow$	$45.93_{\pm 4.60}\downarrow$
	Symbolic CoT	$16.14_{\pm 1.97}\downarrow$	$19.16_{\pm 2.41}\downarrow$	$21.00_{\pm 1.38}\downarrow$	$20.87_{\pm 2.36}\downarrow$	$40.94_{\pm 0.00}\downarrow$	$28.08_{\pm 3.06}\downarrow$	$42.13_{\pm 1.80}\downarrow$
OBQA	Original CoT	$36.73_{\pm 0.76}$	$46.07_{\pm 2.23}$	$48.00_{\pm 1.40}$	$54.87_{\pm 2.02}$	$69.93_{\pm 1.36}$	$63.60_{\pm 2.12}$	$78.80_{\pm 0.80}$
	Least-to-most	$31.40_{\pm 1.74}\downarrow$	$43.33_{\pm 2.02}\downarrow$	$45.53_{\pm 2.39}\downarrow$	$53.20_{\pm 2.50}\downarrow$	$68.27_{\pm 1.03}\downarrow$	$62.80_{\pm 2.23}\downarrow$	$78.33_{\pm 1.62}\downarrow$
	RaR	$\mathbf{40.47}_{\pm 1.68}\uparrow$	$\mathbf{47.47}_{\pm 2.23}\uparrow$	$\mathbf{49.87}_{\pm 1.42}\uparrow$	$\mathbf{56.40}_{\pm 1.97}\uparrow$	$\mathbf{72.73}_{\pm 2.19}\uparrow$	$\mathbf{64.40}_{\pm 2.75}\uparrow$	$\mathbf{82.00}_{\pm 0.20}\uparrow$
	Symbolic CoT	$31.67_{\pm 1.36}\downarrow$	$35.13_{\pm 0.23}\downarrow$	$37.73_{\pm 3.25}\downarrow$	$41.13_{\pm 1.81}\downarrow$	$61.80_{\pm 0.92}\downarrow$	$52.47_{\pm 0.70}\downarrow$	$72.13_{\pm 0.64}\downarrow$

Table 3: Performance of student models with different CoT formats. For each dataset, the best performance is boldfaced, and arrows show that the performance is increased (↑) or decreased (↓) over original CoT.

6 Effects of Format

Beyond granularity, the format of reasoning has been widely believed to influence model performance in prior research. However, SLMs often face limitations in processing complex reasoning structures. This raises a research question: *Do these alternative formats consistently improve student model performance, or are their benefits task-specific and limited?*

Choice of Reasoning Formats In this section, we systematically evaluate the impact of alternative reasoning structures on student model performance. Since student models tend to perform stably at intermediate granularity levels, we let GPT-4o modify the format of the original CoT without changing the granularity (More details can be seen in Appendix D). We compare the original CoT format with three alternative structures:

- **Least-to-most** (Zhou et al. 2023): A reasoning approach that decomposes a complex problem into a sequence of sub problems. Least-to-most excels in systematically breaking down problems into manageable parts to facilitate understanding and solution synthesis.
- **Rephrase and Respond** (RaR) (Deng et al., 2024): A method where questions are rephrased to reduce ambiguity before answering, enabling iterative clarification and improving the LLM's ability to respond accurately to nuanced queries.
- **Symbolic CoT** (Xu et al., 2024a): A reasoning structure that combines symbolic logic and CoT prompting, translating natural language into symbolic expressions for step-by-step logical deduction, enhancing faithfulness and flexibility in problem-solving.

Our results, summarized in Table 3, highlight a clear trend: *the original CoT format often outperforms more complex or modified structures, primarily due to its simplicity and adaptability.* This contrasts with previous findings in LLMs, where these alternative formats frequently yield improvements. For SLMs, however, the added complexity of alternative formats generally increases cognitive load and hardly improve the performance.

Task-Specific Gains While most tasks favor the original CoT format, certain alternative structures offer measurable benefits for specific scenarios. For example, RaR improves commonsense reasoning tasks by reducing ambiguity and enabling iterative clarification. Least-to-most sometimes excels in mathematical reasoning by breaking problems into logical steps or symbolic expressions. [2]

Model-Specific Trends Stronger student models, such as LLaMA 3.2 3B, show improved performance under alternative formats, leveraging structural cues to refine problem-solving processes. However, these improvements are tied to specific tasks and do not generalize across all datasets.

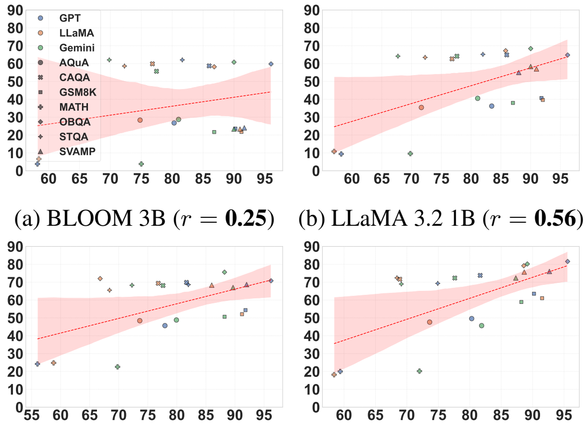
Overall, our findings indicate that while CoT formats can occasionally enhance performance, their benefits are often task-dependent and come at the cost of introducing additional tokens. Moreover,

[2]A possible reason for the suboptimal performance of Symbolic CoT is analyzed with examples in the Appendix G.

SLMs have relatively limited pretraining corpora, which likely contain fewer instances of these reasoning formats. As a result, weaker models struggle to effectively learn and utilize them, making it even harder for CoT format variations to yield consistent improvements. Given these observations, we argue that *adjusting CoT formats alone may not be the most effective approach for improving SLM performance*. This contrasts with the consistent impact of granularity, as highlighted in Section 5, suggesting that *focusing on granularity is a more effective strategy than altering CoT formats*.

Conclusion

While alternative CoT formats sometimes offer some task-specific benefits, the original CoT format from teacher models often remains the most effective for general-purpose SLM training.



(a) BLOOM 3B ($r = \mathbf{0.25}$) (b) LLaMA 3.2 1B ($r = \mathbf{0.56}$)

(c) Gemma 2B ($r = \mathbf{0.55}$) (d) LLaMA 3.2 3B ($r = \mathbf{0.64}$)

Figure 4: Scatter plots of teacher (x-axis) vs. student model accuracy (y-axis) across datasets. GPT refers to GPT-4o, LLaMA refers to LLaMA 3 70B, and Gemini refers to Gemini-1.5-Flash.

7 Effects of Teacher Model

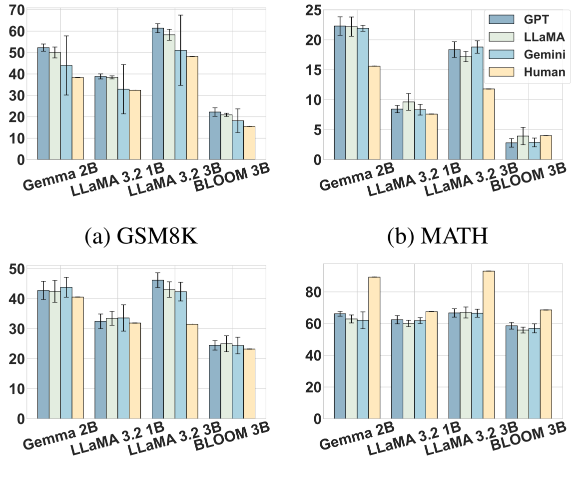
In CoT distillation, teacher models serve as the source of CoT reasoning annotations for training student models. Prior research of KD assumes that teacher models with better performance naturally lead to better student models (Zong et al., 2023). This assumption stems from the belief that higher-performing teachers generate more accurate reasoning steps and answers, which, when distilled into student models, enhance their capabilities. However, this assumption may not hold universally as SLMs might have limited capacity to replicate the reasoning complexity of strong teachers.

In this section, we analyze the performance of student models under four teachers, GPT-4o, LLaMA 3 70B, Gemini-1.5-Flash and the human expert. We aim to determine *whether the choice of teacher affects the ability of student models to effectively distill and replicate CoT reasoning*.

Is Higher Teacher Accuracy Always Better? We first investigate whether a higher teacher accuracy directly translates into improved student performance. As shown in Figure 4, we selected the best-performing student model for each dataset under different teacher models' CoT supervision. It can be observed that points closer to the right side of the x-axis are not always positioned near the top of the y-axis. This indicates that, contrary to intuitive expectations, while a reasonably accurate teacher can effectively impart essential reasoning patterns, excessively high teacher accuracy does not always yield proportional improvements in student accuracy. *Teacher accuracy alone is not the determining factor for student performance*, which aligns with our findings in Section 5.

Moreover, we do not observe a significant preference pattern within the same model family. However, we find that *stronger student models tend to benefit more significantly when trained under the guidance of stronger teacher models*. This suggests that the trade-off between teacher model capability and computational cost should be carefully adjusted based on the target student model's capacity. For simpler tasks, a less advanced teacher model is often sufficient, producing results comparable to those obtained from more powerful, computationally expensive teachers.

Human vs LLM: Task-Specific Effectiveness As seen in Figure 5, in mathematical reasoning tasks, student models achieve higher accuracy when fine-tuned on LLM-generated CoTs compared to human annotations, although the accuracy of the teacher model itself performs poorly on difficult mathematical datasets compared with human-labeled data. Conversely, for commonsense reasoning tasks like StrategyQA, human-annotated CoTs dramatically improve student model performance. This phenomenon arises because LLMs generate *structured and detailed reasoning chains* that closely align with the symbolic and procedural nature of mathematical tasks. In contrast, human-annotated CoTs often lack the rigorous step-by-



(a) GSM8K (b) MATH

(c) AQuA-RAT (d) StrategyQA

Figure 5: Student model performance across different teacher models. Each bar represents the average accuracy of a specific student model trained on CoT from different teacher models.

step structure required for effective mathematical reasoning. However, Human annotations excel at capturing *nuanced contextual understanding, creative inferences, and interpretive reasoning*, which are crucial for tasks involving ambiguous or open-ended questions. These findings underscore *the importance of selecting CoT sources based on task characteristics* rather than assuming a universal superiority of either LLMs or human annotations. Full results can be found in Appendix H.

The Matthew Effect in SLMs We explore the relationship between student model capacity and the benefits gained from CoT distillation, shedding light on the uneven distribution of performance improvements across models of varying capabilities. Figure 6 presents two heatmaps comparing student model performance before and after CoT distillation. The results reveal a *Matthew Effect*: *stronger student models achieve greater performance gains from CoT distillation than weaker models*, demonstrating their potential ability to leverage detailed reasoning steps. This phenomenon aligns with Vygotsky's Zone of Proximal Development (ZPD) (Vygotsky, 1978), where weaker student models have a narrower ZPD, limiting their ability to absorb complex CoT reasoning. If reasoning complexity is too high relative to a model's ZPD, it may fail to extract useful patterns, limiting the effectiveness of CoT distillation. In contrast, stronger models have a wider ZPD, enabling them to integrate and generalize from multi-step reasoning. CoT distillation provides gains on more challenging ones, where their capacity allows them to fully leverage structured reasoning. This highlights the need for adaptive CoT supervision, where reasoning depth is modulated based on the student's ability to process and learn from it.

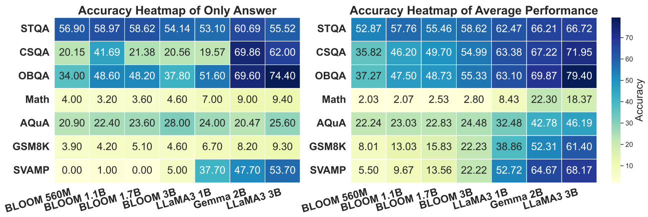


Figure 6: The *Only Answer* heatmap represents the baseline accuracy of student models without reasoning supervision, while the *Average Performance* heatmap shows the average accuracy of student models trained on CoT from ChatGPT-4o.

Conclusion

The assumption that a better teacher always produces a better student does not universally hold for SLMs. Stronger student models benefit more from advanced teacher models. Teacher choice should be task-specific: LLM-generated CoTs improve mathematical reasoning, while human annotations excel in commonsense reasoning.

8 Conclusion

This study systematically examined key factors influencing CoT distillation in SLMs, including teacher selection, granularity, and format. First, We found that finer-grained CoT benefits stronger SLMs, and weaker models perform better with simpler annotations. Then, while CoT format significantly impacts LLMs, its effect on SLMs is more subtle. Importantly, better teacher models do not always yield better students, as the effectiveness of CoT distillation depends on a model's ability to absorb reasoning complexity within its ZPD. Notably, human-annotated CoTs underperform on mathematical tasks but can surpass LLM-generated CoTs in certain commonsense reasoning tasks. Overall, CoT distillation proves more effective for stronger SLMs and complex tasks, emphasizing the need for tailored granularity and teacher selection strategies to optimize reasoning performance in resource-constrained settings.

Limitations

Despite the promising results of our study, several limitations must be acknowledged. First, during data generation and testing, some tasks triggered safety concerns in the models, causing them to refuse to generate CoTs. In these cases, we resorted to directly using the provided answers for fine-tuning, which may have constrained the diversity and quality of the reasoning chains, potentially affecting the distillation outcomes. Second, the ability of teacher models to generate CoTs is inherently tied to their reasoning capabilities. For certain tasks, teacher models were unable to reverse-engineer plausible CoTs from the given answers due to their limited capabilities, resulting in incomplete or suboptimal reasoning chains. Lastly, this study did not focus on exploring novel KD techniques but instead aimed to systematically analyze the effects of existing approaches on CoT granularity, format, and teacher selection. These limitations underscore the need for further research into CoT generation and the development of advanced distillation methods tailored to task-specific requirements.

Ethics Statement

This study adheres to ethical standards in AI research by ensuring that all experiments were conducted using publicly available datasets and pretrained models. During the data generation process, measures were taken to respect model safety constraints, avoiding harmful or inappropriate outputs. While some tasks required bypassing CoT generation due to safety concerns, we ensured that these adjustments did not compromise the ethical integrity of the fine-tuning process. Furthermore, this research aims to optimize reasoning capabilities in SLMs while minimizing computational resources, promoting environmentally sustainable AI practices. We acknowledge that KD techniques may inadvertently propagate biases from teacher models to student models. To mitigate this, we recommend conducting comprehensive evaluations of distillation pipelines to identify and address potential biases before deployment in real-world applications. This work ultimately seeks to advance AI accessibility while prioritizing ethical considerations in model development and deployment.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2024. Rephrase and respond: Let large language models ask better questions for themselves. *Preprint*, arXiv:2311.04205.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Preprint*, arXiv:2101.02235.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

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*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842, Bangkok, Thailand. Association for Computational Linguistics.

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. *Preprint*, arXiv:2001.08361.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

Shanu Kumar, Saish Mendke, Karody Lubna Abdul Rahman, Santosh Kurasa, Parag Agrawal, and Sandipan Dandapat. 2024. Enhancing zero-shot chain of thought prompting via uncertainty-guided strategy selection. *arXiv preprint arXiv:2412.00353*.

Ming Li, Yanhong Li, and Tianyi Zhou. 2024. What happened in llms layers when trained for fast vs. slow thinking: A gradient perspective. *Preprint*, arXiv:2410.23743.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.

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.

William Merrill and Ashish Sabharwal. 2024. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*.

Meta. 2024a. Introducing meta Llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`.

Meta. 2024b. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *Preprint*, arXiv:1809.02789.

Sania Nayab, Giulio Rossolini, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *Preprint*, arXiv:2407.19825.

OpenAI. 2024. Hello GPT-4o. `https://openai.com/index/hello-gpt-4o/`.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *Preprint*, arXiv:2103.07191.

Akshara Prabhakar, Thomas L. Griffiths, and R. Thomas McCoy. 2024. Deciphering the factors influencing the efficacy of chain-of-thought: Probability, memorization, and noisy reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3710–3724, Miami, Florida, USA. Association for Computational Linguistics.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

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.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schoelkopf, and Mrinmaya Sachan. 2023. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 545–561, 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.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024b. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Lev S Vygotsky. 1978. *Mind in society: The development of higher psychological processes*, volume 86. Harvard university press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024a. Faithful logical reasoning via symbolic chain-of-thought. *Preprint*, arXiv:2405.18357.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. *Preprint*, arXiv:2402.13116.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. 2024. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Martin Zong, Zengyu Qiu, Xinzhu Ma, Kunlin Yang, Chunya Liu, Jun Hou, Shuai Yi, and Wanli Ouyang. 2023. Better teacher better student: Dynamic prior knowledge for knowledge distillation. In *The Eleventh International Conference on Learning Representations*.

Appendix

A Overview of Training and Test Datasets

For our experiments, we used three models (Llama3 70B, Gemini-1.5-Flash, GPT-4o) on multiple existing datasets, including mathematical reasoning datasets (SVAMP, GSM8K, AQuA-RAT, MATH) and commonsense reasoning datasets (OpenBookQA, CommonsenseQA, and StrategyQA) to generate CoT outputs. Table 4 shows the overview of the training and test datasets (Yue et al., 2024). Table 5 shows some examples of our datasets.

Training Dataset	Samples		Fields	Human Annotation
	Training	Testing		
SVAMP	700	300	Arithmetic problems	Yes
GSM8K	7.4k	1.3k	Grade-school math	Yes
AQuA-RAT	6.1k	254	Algebraic reasoning, multi-step	Yes
Math	1.3k	500	Pre-Algebra, Algebra, Counting & Probability, Number Theory	Yes
CommonsenseQA	9.7k	1.2k	Commonsense knowledge	Yes
OpenBookQA	4.9k	500	Domain-specific knowledge	No
StrategyQA	2k	290	Multi-step reasoning	Yes

Table 4: Overview of Training and Test Datasets.

Dataset	Problem	Characteristics
SVAMP	There are 87 oranges and 290 bananas in Philip's collection. If the bananas are organized into 2 groups and oranges are organized into 93 groups How big is each group of bananas?	290.0 / 2.0 = 145.0. The answer is 145.0.
GSM8K	Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?	Natalia sold 48/2 = «48/2=24»24 clips in May. Natalia sold 48+24 = «48+24=72»72 clips altogether in April and May. 72
AQuA-RAT	A man can swim in still water at 7.5 km/h, but takes twice as long to swim upstream than downstream. The speed of the stream is? Answer Choices: (A) 3 (B) 2.5 (C) 2.25 (D) 1.5 (E) 4	M = 7.5 S = x DS = 7.5 + x US = 7.5 + x 7.5 + x = (7.5 - x)2 7.5 + x = 15-2x 3x = 7.5 x = 2.5 Answer: C
Math	Find the sum of all positive divisors of 50 that are also divisors of 15.	The positive factors of 50 are 1, 2,5, 10, 25, 50. Of these, only 1 and 5 divide 15. Their sum is 1+5 = 6.
CommonsenseQA	Bill did not abandon the fight, but did what to the enemy? Answer choices: A: arrogate, B: retain, C: embrace, D: smile, E: engage	Bill engaged in a fight with enemy. Other options are not a type of fights one takes with enemy. The answer is E.
StrategyQA	Are more people today related to Genghis Khan than Julius Caesar?	Julius Caesar had three children. Genghis Khan had sixteen children. Modern geneticists have determined that out of every 200 men today has DNA that can be traced to Genghis Khan. The answer is True.

Table 5: Examples of Human Annotation for All Datasets.

B Training setup

Our experiment uses the LLaMA-Factory framework (Zheng et al., 2024) to fine-tune models, and the training parameters are as follows:

Parameter	Value
Learning Rate	3e-5
Num Train Epochs	3
LR Scheduler	Cosine
Max Grad Norm	1.0
Optimizer	AdamW
Template	gemma/alpaca/llama3

Table 6: Configuration for training parameters.

C Different CoT Granularity Dataset Collection

C.1 Workflow

The granularity dataset processing steps are detailed below:

1. 0-Shot Example Generation: The same question is first provided to the three models using a 0-shot prompt. The models generate a single 1-shot example (including both a question and a corresponding output) as the output. This step ensures that the models first generate a baseline example. The generated example serves as a guide for subsequent responses.

2. Input Construction: Each question is provided to the models, along with its corresponding ground-truth answer from the original dataset and the generated 1-shot example (from Step 1). Including the 1-shot example in the input establishes a reference point for the model, enhancing coherence and quality of generated outputs.

3. Generation with Multi-Granularity Outputs: Using the constructed input (original question + ground truth + 1-shot example), all three teacher models are prompted to generate answers at multiple granularity levels ($G = \{g_1, g_2, \ldots, g_6\}$). These levels range from concise summaries to highly detailed, step-by-step reasoning. By solving each question across six levels of granularity, this step systematically evaluates the models' ability to adapt their reasoning to different levels of abstraction.

4. Ranking and Alignment: The generated outputs are sorted to align with the original dataset's order, ensuring consistency and enabling a systematic evaluation of the results. Sorting the generated outputs ensures that the evaluation is systematic and comparable against the original dataset.

Why do we use 1-shot example in the prompt:

We have decided to incorporate a 1-shot example into the prompt instead of using a 0-shot prompt, based on our trial-and-error findings.

Our initial attempt used a forward-generation approach, where we prompted the model to produce the most succinct response and then enrich it level by level. However, we encountered significant challenges with this approach. The model struggled to demonstrate consistent incremental increases in granularity, as the initial requirement for conciseness often constrained its reasoning and led to inaccuracies or incomplete answers. The model's inability to build upon a succinct base made this method unsuitable for achieving the desired level of granularity.

To address this, we reversed the approach by asking the model to provide the most elaborate response, intending to progressively reduce the level of detail in subsequent steps. While this method initially produced more detailed outputs, the responses often lacked sufficient depth and structure to support multiple rounds of granularity reduction. As a result, achieving consistent decreases in detail also proved to be a challenge.

These findings highlighted the need for a more structured and balanced approach. We identified that including a 1-shot example in the prompt could effectively guide the model to produce outputs with consistent and balanced granularity across levels. A well-designed 1-shot example helps the model demonstrate high-quality reasoning even in concise answers, ensuring alignment with task requirements regardless of the level of detail. It also provides a clear reference for maintaining consistency when transitioning between levels of granularity.

In summary, 1-shot prompts strike an effective balance between flexibility and structure, enabling the model to generalize across tasks while maintaining coherence and consistency. This approach significantly enhances the model's ability to generate high-quality training samples with varying levels of reasoning granularity.

As a result, we have decided to generate a 1-shot example to include in the prompt. The first prompt will be used to create the 1-shot example, and the second prompt will leverage it for data generation.

C.2 Prompts

CoT Prompt Template

You are a math teacher. Please think step by step for the following question.

Output the result strictly in the following format. DO NOT generate any other explanations.

The generated answer must be consistent with the given answer.

`Question: "<your question>" Answer: "<original answer>"`

The output format should be as follows:
`"instruction": "<your question>", "output": "<Solution Path> The answer is <answer>"`

Here is the example:
`<example>`

Figure 7: Prompt for generating CoT dataset.

Synthetic 1-shot CoT Example Based on Granularity Prompt Template

You are a math teacher. Please think step by step for the following questions in six different Granularity levels.

Ensure that the explanations become progressively more detailed as the Granularity increases. The difference in the number of words between each Granularity should be as large as possible.

The generated answer must be consistent with the given answer. DO NOT generate any other explanations. Output the result strictly in the following format:

```
"instruction": "<your question>", "output": "<Solution Path>\n The answer is <answer>"
```

Granularity definitions:

- **Level 1**: Provide the most essential steps to reach the answer, minimizing explanations and focusing on the direct path to the solution.
- **Level 2**: Provide the essential steps required to reach the answer, including some intermediate calculations. It should be more detailed than level 1 but shorter than level 3.
- **Level 3**: Provide a detailed breakdown that includes all necessary calculations and explanations but shorter and less detailed than level 4. Ensure it is more detailed than level 2.
- **Level 4**: Provide a very detailed breakdown that includes all necessary calculations and explanations but avoids extra clarifications that would belong to level 5. It should be more detailed than level 3.
- **Level 5**: Provide an extremely detailed breakdown, covering almost all aspects and sub-steps, but shorter and less detailed than level 6. It should include additional clarifications and checks beyond level 4.
- **Level 6**: Provide the most detailed and comprehensive breakdown, covering all possible aspects and sub-steps. Ensure this explanation is significantly longer and more detailed than level 5.

The output format should be as follows:

```
{ "Granularity": "level 1", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
{ "Granularity": "level 2", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
{ "Granularity": "level 3", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
{ "Granularity": "level 4", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
{ "Granularity": "level 5", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
{ "Granularity": "level 6", "instruction": "<your question>", "output": "Explanation: <Explanation>\n The answer is <answer>" }
```

Figure 8: Synthetic 1-shot CoT prompt example based on granularity levels, demonstrating progressively detailed reasoning steps from Level 1 to Level 6 explanations.

C.3 Granularity case study

Question: At the beginning of every period of British Literature, Mrs. Crabapple picks a random student to receive a crabapple as a gift. Given that there are 11 students in her class and her class meets four times a week, how many different sequences of crabapple recipients are possible in a week?

C.3.1 GPT-4o

Granularity	Generated Output
Level 1	1. Number of students = 11 2. Class meets per week = 4 times 3. Each day, 1 student can receive a crabapple 4. Choices per day = 11 5. Total choices = $11^4 = 14641$ The answer is 14,641
Level 2	1. Each period is an independent random event where 1 of 11 students is chosen. 2. There are 4 periods per week. 3. Each of the 4 periods has 11 possible choices. 4. The total number of sequences is 11^4. 5. Calculate 11^4: $11 \times 11 = 121, 121 \times 11 = 1331, 1331 \times 11 = 14641$ The answer is 14,641
Level 3	1. Each time Mrs. Crabapple picks a student, she has 11 choices. 2. Since the class meets four times a week, there are four independent selections, each with 11 choices. 3. The number of different sequences of crabapple recipients is calculated by raising the number of choices to the power corresponding to the number of selections: 11^4. 4. Calculate this value: $11 \times 11 \times 11 \times 11 = 14,641$. The answer is 14,641
Level 4	1. Each class session, Mrs. Crabapple picks one of the 11 students. 2. Multiply the number of choices per session: 11 choices per session. 3. Since the class meets four times a week, we need to raise the number of choices to the power of the number of sessions: 11^4. 4. Calculate 11^4: $11 \times 11 \times 11 \times 11$. 5. Perform the multiplication step-by-step: $11 \times 11 = 121, 121 \times 11 = 1331, 1331 \times 11 = 14,641$. 6. Review all steps for consistency: 11 choices per session, 4 sessions total, 11^4 computation confirms 14,641 possibilities. The answer is 14,641
Level 5	1. Identify key variables: 11 students, 4 class meetings. 2. Recognize the problem's nature: finding the number of different sequences (order matters) of recipients over 4 periods. 3. Analyze each period's choice: Each of the 4 periods can be filled by any one of the 11 students. 4. Calculate total sequences using the multiplication principle: Each of 4 periods has 11 possible choices. 5. Compute total different sequences: $11 \times 11 \times 11 \times 11 = 11^4$. 6. Evaluate numerical result: $11^4 = 11 \times 11 \times 11 \times 11 = 14,641$. The answer is 14,641
Level 6	1. Determine possible choices per class: Mrs. Crabapple can choose one out of 11 students each time. 2. Confirm weekly meetings: The class has four periods a week. 3. Establish sequence combination formula: For each period, there are 11 choices, resulting in 11 choices for the first period, 11 for the second, 11 for the third, and 11 for the fourth. 4. Calculate total sequences: Multiply available choices for each period to find total combinations: $11 \times 11 \times 11 \times 11 = 11^4$. 5. Compute final result: Raising 11 to the power of 4, we get $11^4 = 14,641$ sequences. The answer is 14,641

Table 7: Comparison of Different CoT granularity for the same problem generated by GPT-4o.

C.3.2 LLaMA 3 70B

Granularity	Generated Output
Level 1	The number of possible recipients for each class is 11. Since class meets 4 times a week, the total number of sequences is 11*11*11*11 = 11^4 = 14641. The answer is 14,641
Level 2	First, we need to find the number of possible recipients for each class period, which is 11. Then, we need to find the number of possible sequences for the week, which is the number of possible recipients for each class period raised to the power of the number of class periods in a week: 11^4 = 14641. The answer is 14,641
Level 3	Step 1: Determine the number of students in the class. There are 11 students in Mrs. Crabappleś British Literature class. Step 2: Determine the number of times the class meets in a week. The class meets 4 times a week. Step 3: Calculate the number of different sequences of crabapple recipients possible in a week. Since there are 11 students and the class meets 4 times a week, we can calculate the number of different sequences by raising 11 to the power of 4, which represents the number of choices for each of the 4 days. 11^4 = 14641 The answer is 14,641
Level 4	Step 1: Determine the number of students in the class. There are 11 students in Mrs. Crabappleś British Literature class. Step 2: Determine the number of times the class meets in a week. The class meets 4 times a week. Step 3: Calculate the number of different sequences of crabapple recipients possible in a week. Since Mrs. Crabapple picks a random student each time, the number of sequences is equal to the number of permutations of 11 students taken 4 at a time, allowing for repetition (since the same student can be picked more than once). This is calculated as 11^4. Step 4: Calculate 11^4. 11^4 = 11 * 11 * 11 * 11 = 14641. The answer is 14,641
Level 5	To solve this problem, we need to use the concept of permutations. Since Mrs. Crabapple picks a random student each time, the order of selection matters. We have 11 students to choose from for each of the 4 periods in a week. For the first period, there are 11 possible choices. For the second period, there are again 11 possible choices, and so on. Since the choices are independent of each other, we can multiply the number of choices for each period to get the total number of different sequences. This gives us (11 times 11 times 11 times 11 = 11^4 = 14641). Therefore, there are 14641 different sequences of crabapple recipients possible in a week. The answer is 14,641
Level 6	To solve this problem, we need to use the concept of permutations. Since Mrs. Crabapple picks a random student to receive a crabapple as a gift in each class, and there are 11 students in her class, there are 11 possible choices for the first class.For the second class, there are also 11 possible choices, and so on. Since her class meets four times a week, we need to find the total number of different sequences of crabapple recipients for four classes. This can be calculated by multiplying the number of choices for each class together. So, the total number of different sequences is (11 times 11 times 11 times 11 = 11^4). To calculate (11^4), we can multiply 11 by itself four times. This gives us 11 times 11 = 121, 121 times 11 = 1331, and 1331 times 11 = 14641. Therefore, the total number of different sequences of crabapple recipients possible in a week is 14641. The answer is 14,641

Table 8: Comparison of Different CoT granularity for the same problem generated by LLaMA 3 70B.

C.3.3 Gemini-1.5-Flash

Granularity	Generated Output
Level 1	There are 11 choices for the recipient of the crabapple in each of the four class periods. The number of possible sequences is therefore 11 * 11 * 11 * 11 = 11^4. 11^4 = 14641. The answer is 14,641
Level 2	There are 11 students in the class. Mrs. Crabapple picks a student each of the four class periods. The number of choices for the first period is 11. The number of choices for the second period is 11. The number of choices for the third period is 11. The number of choices for the fourth period is 11. To find the total number of different sequences, we multiply the number of choices for each period together: 11 * 11 * 11 * 11 = 11^4 = 14641. The answer is 14,641
Level 3	There are 11 students in the class. Mrs. Crabapple picks a student each of the four class periods. The number of choices for the first period is 11. The number of choices for the second period is also 11 (since the same student can receive a crabapple multiple times). Similarly, there are 11 choices for the third period and 11 choices for the fourth period. To find the total number of different sequences, we multiply the number of choices for each period together. This is 11 times 11 times 11 times 11 = 11^4. Calculating 11^4 gives us 14641. The answer is 14,641
Level 4	There are 11 students in the class, and the class meets 4 times a week. Each time Mrs. Crabapple picks a student, there are 11 choices. Since there are four periods, we multiply the number of choices for each period together to find the total number of different sequences. This is 11 * 11 * 11 * 11 = 11^4. Calculating 11^4: 11 * 11 = 121; 121 * 11 = 1331; 1331 * 11 = 14641. The answer is 14,641
Level 5	There are 11 students in the class, and the class meets four times a week. Each time the class meets, Mrs. Crabapple picks one student to receive a crabapple. The number of choices for the first period is 11. The number of choices for the second period is also 11 (since the same student can receive a crabapple again). Similarly, there are 11 choices for the third period and 11 choices for the fourth period. To find the total number of different sequences of crabapple recipients, we multiply the number of choices for each period together. This is because each choice is independent of the others. Therefore, the total number of sequences is 11 * 11 * 11 * 11 = 11^4. Calculating 11^4: 11 * 11 = 121; 121 * 11 = 1331; 1331 * 11 = 14641. The answer is 14,641
Level 6	There are 11 students in the class, and the class meets four times a week. Each time the class meets, Mrs. Crabapple picks one student to receive a crabapple. The number of ways to choose a student for the first period is 11. The number of ways to choose a student for the second period is also 11. Similarly, there are 11 ways to choose a student for the third period and 11 ways to choose a student for the fourth period. Since these choices are independent, we multiply the number of choices for each period together to find the total number of different sequences of crabapple recipients in a week. This is 11 * 11 * 11 * 11 = 11^4. Calculating 11^4, we get 14641. Therefore, there are 14,641 different sequences of crabapple recipients possible in a week. The answer is 14,641

Table 9: Comparison of Different CoT granularity for the same problem generated by Gemini-1.5-Flash.

D Different CoT Format Dataset Collection

D.1 Workflow

The dataset with format processing steps are detailed below:

1. 0-Shot Example Generation: The same question is first provided to ChatGPT using a 0-shot prompt in three formats. ChatGPT generates a single 1-shot example for each format (including both the question and a corresponding output in the specific format). This step ensures that ChatGPT establishes a baseline example for each format. These generated examples serve as templates for subsequent responses, ensuring consistency in style and logic.

2. Input Construction: Each question is then re-input into ChatGPT, along with: - Its corresponding original ChatGPT-generated outputs. - The 1-shot examples generated in Step 1 for all three reasoning formats.

Including the 1-shot examples in the input serves as explicit format demonstrations, guiding ChatGPT to generate outputs that align with the desired styles. This process improves the coherence and quality of the resulting outputs.

3. Multi-Format Output Generation: Using the constructed input (original question + original outputs + 1-shot examples), ChatGPT generates reformatted outputs for each question across three reasoning formats while preserving the original logic: Least-to-most, RaR and SymbolicCoT.

4. Ranking and Alignment: The reformatted outputs are then sorted to align with the original dataset's order. This step ensures consistency and enables systematic evaluation. Sorting the outputs guarantees that the evaluation is both structured and comparable across different reasoning formats and the original dataset.

D.2 Prompt

Symbolic CoT Prompt Template

Please rewrite the output by following the Symbolic CoT (SymbCoT) reasoning to solve the given question step-by-step. You can ONLY change the format but not the original steps. In your rewrite, translate the question's context into symbolic logic format, identifying key variables and relationships. Ensure to use logical symbols such as $\exists$ (exists), $\forall$ (for all), $\wedge$ (and), $\vee$ (or) and $\implies$ (implies), etc., to represent relationships between variables.
You should use symbolic thinking steps in the output. The generated output must follow this specific structure and include logical symbols. Output the result strictly in the following format. DO NOT generate any other explanations. The generated answer must be consistent with the given answer. After modification, you must add `The answer is <answer>` at the end.
Here is the original output:

{`instruction: <question>`, `output: <solution_path>`}

The output format should be as follows:

{`instruction: <question>`, `output: <SymbCoT Solution Path> The answer is <answer>`}

Here is the example: `<example>`

Figure 9: Prompt for generating Symbolic Chain-of-Thought (SymbCoT) reasoning, requiring the transformation of problem contexts into symbolic logic representations using logical operators ($\exists$, $\forall$, $\wedge$, $\vee$ and $\implies$).

Rephrase and Respond (RaR) CoT Prompt Template

Please rewrite the output and answer them individually. You can ONLY change the format but not the original steps. The generated answer must be consistent with the given answer. Output the result strictly in the following format. DO NOT generate any other explanations.
After modification, you must add `The answer is <answer>` at the end.
Here is the original output:

```
{instruction: <question>, output: <solution_path>}
```

Rephrase and expand the given question, and then respond carefully.
The output format should be as follows:

```
{instruction: <question>,
  output: <Rephrase and expand the given question>
  <solution path> The answer is <answer>}
```

Here is the example: `<example>`

Figure 10: Prompt for Rephrase and Respond (RaR), which restructures the question while maintaining the original reasoning steps.

Least-to-Most CoT Prompt Template

Please rewrite the output by generating sub-questions and answer them individually. You can ONLY change the format but not the original steps.
The generated answer must be consistent with the given answer. Output the result strictly in the following format. DO NOT generate any other explanations.
After modification, you must add `The answer is <answer>` at the end.
Here is the original output:

```
{instruction: <question>, output: <solution_path>}
```

The output format should be as follows:

```
{instruction: <question>,
  output: <Subquestion_1><Explanation_1>
  <Subquestion_2><Explanation_2>... The answer is <answer>}
```

Here is the example: `<example>`

Figure 11: Prompt for Least-to-Most CoT, which restructures the solution by generating and answering sub-questions step-by-step while preserving the original reasoning process.

D.3 Format Case Study

Different CoT Format Examples

Original CoT:
Bert currently has 20 kangaroos. Kameron has 100 kangaroos. The difference in the number of kangaroos is 100 - 20 = 80 kangaroos. Bert buys kangaroos at a rate of 2 kangaroos per day. Therefore, the number of days required for Bert to have the same number of kangaroos as Kameron currently has is 80 / 2 = 40 days. The answer is 40.

Symbolic CoT:
1. Let $K_0 = 100\ \exists K_0$. (Kameron has 100 kangaroos).
2. Let $B_0 = 20\ \exists B_0$. (Bert has 20 kangaroos).
3. The difference in kangaroos, $D = K_0 - B_0\ \exists D$:

$$D = 100 - 20 = 80$$

4. Rate of buying kangaroos per day, $r = 2\ \forall r$:

$$r = 2$$

5. Number of days required, $N = \frac{D}{r}\ \exists N$:

$$N = \frac{80}{2} = 40$$

The answer is 40.

Rephrase and Respond:
On Kameron's large farm, there are currently 100 kangaroos, whereas Bert's farm has only 20 kangaroos. Bert aims to increase his kangaroo population, purchasing 2 new kangaroos each day, until he reaches the current number of kangaroos on Kameron's farm.
Calculate the number of days Bert will need to achieve this goal. Bert currently has 20 kangaroos. Kameron has 100 kangaroos. The difference in the number of kangaroos is 100 - 20 = 80 kangaroos. Bert buys kangaroos at a rate of 2 kangaroos per day. Therefore, the number of days required for Bert to have the same number of kangaroos as Kameron currently has is 80 / 2 = 40 days. The answer is 40.

LEAST-TO-MOST:
How many kangaroos does Bert currently have? Bert currently has 20 kangaroos. How many kangaroos does Kameron currently have? Kameron has 100 kangaroos. What is the difference in the number of kangaroos between Bert and Kameron? The difference in the number of kangaroos is 100 - 20 = 80 kangaroos. What is the rate at which Bert buys kangaroos per day? Bert buys kangaroos at a rate of 2 kangaroos per day. How many days will it take for Bert to have the same number of kangaroos as Kameron? The number of days required for Bert to have the same number of kangaroos as Kameron currently has is 80 / 2 = 40 days.
The answer is 40.

Figure 12: Comparison of Different CoT Formats for the same problem: **Kameron has 100 kangaroos on his large farm; Bert has 20 kangaroos on his farm. In how many more days will Bert have the same number of kangaroos as Kameron does now if he buys kangaroos at the same rate of 2 new kangaroos per day?**

E Whole Results of Granularity Experiments

Dataset	Only Answer	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
		Gemma 2B Performance					
SVAMP	47.70	59$_{\pm4.58}$	64.33$_{\pm0.00}$	65.22$_{\pm0.69}$	65.89$_{\pm0.38}$	**67.11**$_{\pm1.35}$ $^{\uparrow13.74\%}$	66.89$_{\pm1.02}$
GSM8K	8.20	49.66$_{\pm0.27}$	52.36$_{\pm0.98}$	53.37$_{\pm0.33}$	52.69$_{\pm0.13}$	53.42$_{\pm0.83}$	**53.45**$_{\pm1.48}$ $^{\uparrow7.63\%}$
AQuA-RAT	20.47	40.68$_{\pm1.27}$	42.91$_{\pm1.42}$	43.7$_{\pm2.58}$	39.9$_{\pm1.49}$	**44.88**$_{\pm0.79}$ $^{\uparrow12.48\%}$	44.49$_{\pm2.36}$
MATH	9.00	23.4$_{\pm1.06}$	21.53$_{\pm2.16}$	**24.4**$_{\pm0.20}$ $^{\uparrow16.19\%}$	21.93$_{\pm0.42}$	23.0$_{\pm1.22}$	21.0$_{\pm0.69}$
CSQA	**69.86**	67.38$_{\pm0.82}$	67.98$_{\pm0.37}$	68.74$_{\pm1.30}$	66.75$_{\pm0.53}$	67.54$_{\pm0.47}$	66.01$_{\pm1.50}$
OBQA	69.60	71.53$_{\pm1.94}$	69.93$_{\pm0.90}$	69.93$_{\pm1.36}$	68.33$_{\pm1.27}$	**72.00**$_{\pm1.64}$ $^{\uparrow5.37\%}$	70.13$_{\pm1.62}$
STQA	60.69	**67.59**$_{\pm1.04}$ $^{\uparrow7.11\%}$	63.1$_{\pm1.79}$	64.6$_{\pm1.56}$	63.45$_{\pm1.24}$	65.75$_{\pm1.77}$	64.14$_{\pm1.58}$
		LLaMA 3.2 1B Performance					
SVAMP	37.70	52.67$_{\pm2.52}$	52.11$_{\pm2.27}$	52.78$_{\pm0.84}$	**53.44**$_{\pm1.17}$ $^{\uparrow2.55\%}$	52.44$_{\pm1.71}$	52.78$_{\pm3.5}$
GSM8K	6.70	36.8$_{\pm0.77}$	39.73$_{\pm0.67}$	**40.08**$_{\pm0.98}$ $^{\uparrow8.91\%}$	39.32$_{\pm1.33}$	39.58$_{\pm1.04}$	38.54$_{\pm0.96}$
AQuA-RAT	24.00	**34.12**$_{\pm1.82}$ $^{\uparrow12.57\%}$	30.31$_{\pm1.42}$	30.58$_{\pm1.2}$	31.23$_{\pm2.02}$	33.2$_{\pm2.17}$	30.45$_{\pm0.91}$
MATH	7.00	**8.87**$_{\pm0.92}$ $^{\uparrow11.85\%}$	8.07$_{\pm1.10}$	8.4$_{\pm0.35}$	8.27$_{\pm0.12}$	7.93$_{\pm1.14}$	8.33$_{\pm0.12}$
CSQA	19.57	**64.48**$_{\pm1.20}$ $^{\uparrow5.39\%}$	63.25$_{\pm0.34}$	62.9$_{\pm1.21}$	61.94$_{\pm1.58}$	62.68$_{\pm1.18}$	61.18$_{\pm0.41}$
OBQA	51.60	**64.4**$_{\pm1.25}$ $^{\uparrow2.01\%}$	63.73$_{\pm1.36}$	63.6$_{\pm2.12}$	63.6$_{\pm1.25}$	63.27$_{\pm1.36}$	63.13$_{\pm0.70}$
STQA	53.10	63.33$_{\pm1.59}$	60.11$_{\pm1.30}$	63.56$_{\pm1.59}$	**64.14**$_{\pm1.50}$ $^{\uparrow6.70\%}$	61.84$_{\pm1.44}$	63.33$_{\pm2.30}$
		LLaMA 3.2 3B Performance					
SVAMP	53.70	69$_{\pm3.61}$	65.89$_{\pm3.53}$	68.33$_{\pm1.86}$	68.11$_{\pm3.27}$	69.78$_{\pm1.07}$	**74.33**$_{\pm1.45}$ $^{\uparrow12.81\%}$
GSM8K	9.30	59.59$_{\pm1.59}$	62.29$_{\pm1.41}$	62.57$_{\pm0.80}$	**63.48**$_{\pm0.16}$ $^{\uparrow6.53\%}$	62.29$_{\pm1.22}$	60.98$_{\pm0.31}$
AQuA-RAT	19.60	44.36$_{\pm2.31}$	44.88$_{\pm2.19}$	45.01$_{\pm2.37}$	46.19$_{\pm3.94}$	**47.24**$_{\pm4.77}$ $^{\uparrow6.49\%}$	46.33$_{\pm3.01}$
MATH	9.40	19.07$_{\pm0.90}$	19.6$_{\pm1.06}$	19.73$_{\pm1.72}$	**20.27**$_{\pm1.42}$ $^{\uparrow11.37\%}$	19.93$_{\pm2.20}$	18.2$_{\pm1.64}$
CSQA	62.00	72.62$_{\pm0.82}$	70.71$_{\pm0.70}$	**74.12**$_{\pm0.50}$ $^{\uparrow4.82\%}$	71.75$_{\pm0.62}$	71.17$_{\pm1.03}$	71.44$_{\pm0.90}$
OBQA	74.40	79.33$_{\pm0.42}$	79.73$_{\pm0.70}$	78.8$_{\pm0.80}$	77.8$_{\pm0.92}$	79.27$_{\pm2.04}$	**80.2**$_{\pm1.78}$ $^{\uparrow3.08\%}$
STQA	55.52	66.44$_{\pm1.55}$	62.76$_{\pm2.82}$	67.47$_{\pm1.44}$	66.78$_{\pm1.39}$	63.91$_{\pm2.79}$	**68.62**$_{\pm1.20}$ $^{\uparrow9.34\%}$
		BLOOM 560M Performance					
SVAMP	0.00	5.11$_{\pm0.19}$	4.56$_{\pm0.77}$	**6.67**$_{\pm1.33}$ $^{\uparrow46.27\%}$	6.56$_{\pm1.90}$	5.56$_{\pm2.41}$	5.11$_{\pm0.69}$
GSM8K	3.90	7.25$_{\pm0.77}$	8.11$_{\pm0.08}$	**8.47**$_{\pm0.12}$ $^{\uparrow16.83\%}$	7.73$_{\pm0.82}$	8.19$_{\pm0.27}$	8.11$_{\pm0.77}$
AQuA-RAT	20.90	**22.05**$_{\pm3.43}$ $^{\uparrow18.29\%}$	20.6$_{\pm2.56}$	21.13$_{\pm4.55}$	21.52$_{\pm1.59}$	18.64$_{\pm1.98}$	19.69$_{\pm1.04}$
MATH	**4.00**	2.6$_{\pm0.53}$	2.33$_{\pm1.01}$	2.13$_{\pm0.61}$	2.13$_{\pm1.15}$	1.67$_{\pm0.42}$	1.87$_{\pm0.50}$
CSQA	20.15	37.76$_{\pm1.29}$	**37.95**$_{\pm0.29}$ $^{\uparrow18.02\%}$	37.84$_{\pm0.74}$	33.99$_{\pm0.83}$	33.96$_{\pm1.57}$	31.89$_{\pm1.23}$
OBQA	34.00	**41.07**$_{\pm0.95}$ $^{\uparrow18.04\%}$	38.27$_{\pm0.42}$	36.73$_{\pm0.76}$	35.73$_{\pm1.40}$	34.8$_{\pm3.64}$	36.27$_{\pm2.04}$
STQA	**56.90**	53.1$_{\pm1.25}$	52.18$_{\pm1.30}$	52.07$_{\pm0.35}$	52.99$_{\pm0.40}$	53.45$_{\pm1.58}$	54.83$_{\pm1.73}$
		BLOOM 1.1B Performance					
SVAMP	1.00	7.22$_{\pm0.69}$	9.11$_{\pm0.51}$	**13.00**$_{\pm1.86}$ $^{\uparrow80.06\%}$	11.33$_{\pm1.20}$	10.67$_{\pm1.00}$	9.89$_{\pm1.64}$
GSM8K	4.20	11.32$_{\pm0.16}$	14.3$_{\pm0.37}$	**14.33**$_{\pm0.79}$ $^{\uparrow26.59\%}$	14.28$_{\pm0.66}$	13.09$_{\pm0.83}$	11.78$_{\pm0.77}$
AQuA-RAT	22.40	21.39$_{\pm2.27}$	21.78$_{\pm1.20}$	21.92$_{\pm1.94}$	21.78$_{\pm1.27}$	21.92$_{\pm3.66}$	**23.36**$_{\pm3.16}$ $^{\uparrow9.21\%}$
MATH	**3.20**	2.33$_{\pm0.31}$	2.73$_{\pm0.31}$	2.4$_{\pm0.40}$	2.07$_{\pm0.58}$	2.2$_{\pm1.22}$	2.87$_{\pm0.76}$
CSQA	41.69	**49.14**$_{\pm2.21}$ $^{\uparrow18.35\%}$	48.21$_{\pm2.35}$	48.57$_{\pm0.36}$	43.87$_{\pm0.82}$	45.37$_{\pm0.79}$	41.52$_{\pm0.91}$
OBQA	**48.60**	48.4$_{\pm3.30}$	47.27$_{\pm1.81}$	46.07$_{\pm2.23}$	46.00$_{\pm2.09}$	46.2$_{\pm2.62}$	46.4$_{\pm2.96}$
STQA	58.97	57.93$_{\pm2.39}$	**59.65**$_{\pm2.48}$ $^{\uparrow6.35\%}$	58.05$_{\pm2.08}$	58.97$_{\pm2.41}$	56.09$_{\pm2.22}$	57.36$_{\pm1.21}$
		BLOOM 1.7B Performance					
SVAMP	0.00	10$_{\pm1.67}$	9.44$_{\pm1.02}$	**17.11**$_{\pm1.17}$ $^{\uparrow71.1\%}$	16.44$_{\pm1.39}$	16.56$_{\pm0.51}$	11.33$_{\pm2.91}$
GSM8K	5.10	13.12$_{\pm0.33}$	17.11$_{\pm0.62}$	16.68$_{\pm0.46}$	**17.89**$_{\pm1.25}$ $^{\uparrow36.36\%}$	16.86$_{\pm1.25}$	15.21$_{\pm0.12}$
AQuA-RAT	23.60	**25.07**$_{\pm0.91}$ $^{\uparrow14.37\%}$	23.88$_{\pm0.45}$	22.05$_{\pm5.46}$	22.7$_{\pm0.82}$	22.31$_{\pm1.38}$	21.92$_{\pm2.17}$
MATH	**3.60**	2.87$_{\pm0.50}$	1.9$_{\pm0.14}$	2.8$_{\pm0.53}$	1.87$_{\pm0.12}$	2.0$_{\pm0.60}$	2.6$_{\pm0.53}$
CSQA	21.38	**53.37**$_{\pm2.21}$ $^{\uparrow15.12\%}$	51.24$_{\pm0.90}$	51.52$_{\pm0.62}$	49.06$_{\pm0.59}$	46.79$_{\pm0.68}$	46.36$_{\pm0.59}$
OBQA	48.20	49.93$_{\pm3.00}$	**50.93**$_{\pm0.99}$ $^{\uparrow8.99\%}$	48.0$_{\pm1.40}$	47.6$_{\pm1.00}$	47.67$_{\pm2.50}$	46.73$_{\pm2.37}$
STQA	**58.62**	54.94$_{\pm1.55}$	56.44$_{\pm2.49}$	58.28$_{\pm0.91}$	57.24$_{\pm2.69}$	56.89$_{\pm5.10}$	56.44$_{\pm1.90}$
		BLOOM 3B Performance					
SVAMP	5.00	15.44$_{\pm0.51}$	23.67$_{\pm0.00}$	23.11$_{\pm1.26}$	**24.00**$_{\pm0.67}$ $^{\uparrow55.44\%}$	22.22$_{\pm0.69}$	22.22$_{\pm1.02}$
GSM8K	4.60	18.2$_{\pm0.57}$	22.34$_{\pm1.14}$	**23.81**$_{\pm0.65}$ $^{\uparrow30.82\%}$	22.57$_{\pm0.88}$	22.47$_{\pm0.86}$	20.85$_{\pm0.15}$
AQuA-RAT	**28.00**	24.67$_{\pm0.82}$	24.41$_{\pm1.72}$	20.34$_{\pm1.82}$	26.90$_{\pm2.41}$	25.85$_{\pm0.45}$	24.28$_{\pm2.17}$
MATH	**4.60**	3.2$_{\pm1.04}$	2.8$_{\pm0.40}$	2.33$_{\pm0.61}$	2.73$_{\pm0.23}$	3.53$_{\pm0.50}$	2.8$_{\pm0.20}$
CSQA	20.56	**57.44**$_{\pm1.12}$ $^{\uparrow11.38\%}$	55.23$_{\pm1.47}$	55.42$_{\pm0.64}$	53.65$_{\pm0.22}$	52.96$_{\pm0.29}$	51.57$_{\pm1.15}$
OBQA	37.80	57.2$_{\pm1.59}$	52.33$_{\pm0.61}$	54.87$_{\pm2.02}$	54.6$_{\pm1.04}$	**57.47**$_{\pm2.64}$ $^{\uparrow9.82\%}$	52.93$_{\pm1.81}$
STQA	54.14	58.85$_{\pm1.74}$	**61.04**$_{\pm3.06}$ $^{\uparrow3.72\%}$	60.58$_{\pm3.09}$	59.89$_{\pm3.13}$	59.19$_{\pm1.21}$	59.08$_{\pm2.87}$

Table 10: Performance of various models at six granularity levels, including standard deviation ($\pm$*std*). The best performance is boldfaced, and red text shows the relative improvement (%) for the highest vs. lowest performance in six levels. *Only Answer*: Student models are fine-tuned to directly predict answers without CoT.

F Padding Procedure for Matched-Length CoT Variants

The following algorithm outlines the process of constructing matched-length CoT variants $\mathcal{D}'_g$, ensuring that sequences from lower granularity levels are padded to match the length of higher granularity sequences. This process is designed to isolate the impact of granularity from sequence length during evaluation.

Algorithm 1: Dynamic Padding for Matched-Length CoT Variants

```
Input: D_1, D_5 (CoT outputs for levels 1, 5)
Output: D'_1: Padded D_1 matching D_5
Load D_1, D_5, ensure lengths match;
foreach (x_1, x_5) ∈ (D_1, D_5) do
    Tokenize x_1, x_5 to get lengths len(T_1) and len(T_5);
    if len(T_1) < len(T_5) then
        Compute repeat_factor = len(T_5)/len(T_1);
        Repeat and truncate T_1 to match len(T_5), decode to x_1^tmp;
        if "The answer is" in x_1 then
            Split x_1 as [prefix, suffix] at "The answer is";
            x'_1 = prefix + x_1^tmp + "The answer is" + suffix;
        else
            x'_1 = x_1^tmp;
        Append x'_1 to D'_1;
Save D'_1 and log statistics;
```

G Analysis of Symbolic CoT Performance in CoT distillation

To investigate the suboptimal performance of Symbolic CoT, we conducted a case study.

Question: janet's ducks lay 16 eggs per day. she eats three for breakfast every morning and bakes muffins for her friends every day with four. she sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?

BLOOM 3B Output

Let:

$$
\begin{aligned}
D &= 16 \quad \text{(eggs laid per day)}\\
E &= 3 \quad \text{(eggs eaten per day)}\\
B &= 3 \quad \text{(breakfasts per day)}\\
M &= 4 \quad \text{(muffins per day)}\\
S &= D - E = 16 - 3 = 13 \quad \text{(sold eggs per day)}\\
R &= \text{remaining eggs after breakfasts and muffins}\\
T &= \text{total daily earnings}
\end{aligned}
$$

1. $\forall E : E = 3$ (Janet eats 3 eggs per day)
2. $\forall B : B = 3$ (Janet bakes muffins for her friends every day)
3. $\forall M : M = 4$ (Janet sells the remaining eggs at the farmers' market daily for $2 per fresh duck egg)
4. $\forall R : R = D - E - M = 16 - 3 - 4 = 9$ (remaining eggs after breakfasts and muffins)
5. $\forall T : T = R \times 2 \Rightarrow T = 13 \times 2 = 26$ (total daily earnings)

The answer is 26.

Gemma 2B Output

Let:

$$
\begin{aligned}
E &= 16 \quad \text{(total eggs laid per day by ducks)} \\
B &= 3 \quad \text{(eggs eaten for breakfast per day)} \\
M &= 4 \quad \text{(eggs baked into muffins per day)} \\
R &= \text{remaining eggs after breakfast and muffins} \\
S &= \text{eggs sold at farmers' market per day} \\
P &= 2 \quad \text{(\$ per egg at farmers' market)} \\
T &= \text{total earnings per day at farmers' market}
\end{aligned}
$$

1. $\forall E : E = 16$ (total eggs laid per day)
2. $\forall B : B = 3$ (Janet eats 3 eggs for breakfast)
3. $\forall M : M = 4$ (Janet bakes 4 muffins daily)
4. Remaining eggs after breakfast and muffins:

$$R = E - B - M \Rightarrow R = 16 - 3 - 4 = 9$$

5. $\forall S : S = R \Rightarrow S = 9$ (Janet sells the remaining eggs)
6. Total earnings:

$$T = S \times P \Rightarrow T = 9 \times 2 = 18$$

The answer is 18.

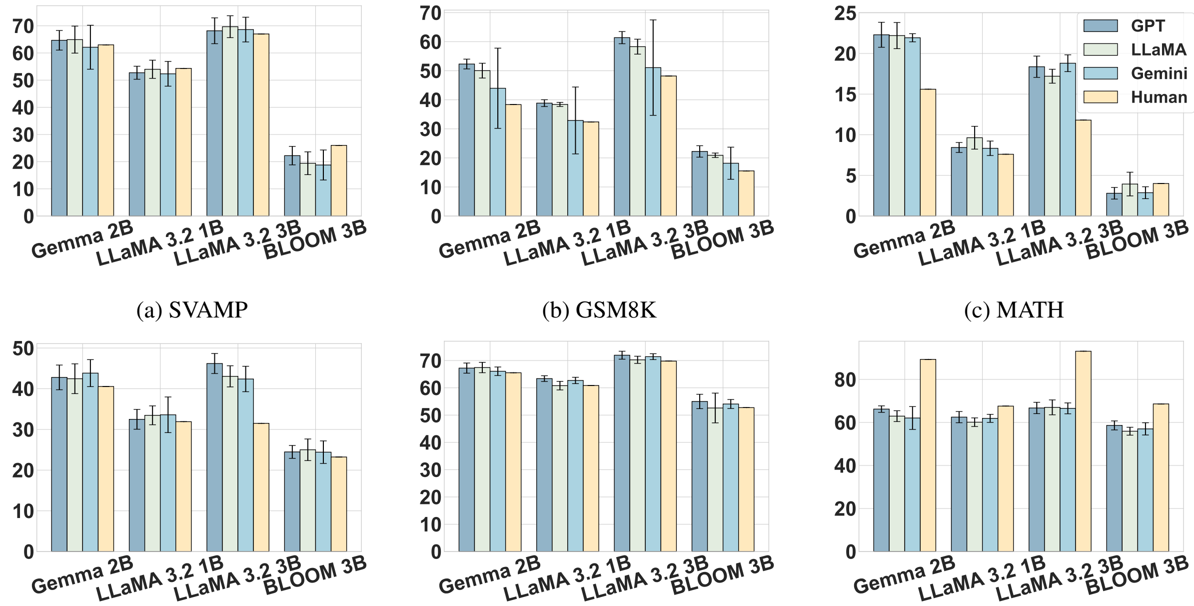
The error in **BLOOM 3B** arises from unnecessary variable assignments and mismanagement of numerical values. While it correctly defines and calculates $R = 9$ (the remaining eggs), it later incorrectly substitutes $R = 13$ into the final computation, leading to an incorrect total earnings prediction of **$26** instead of **$18**. The symbolic expressions themselves are structurally correct, but the model fails to maintain numerical consistency throughout the reasoning process. This suggests that Symbolic CoT lacks a robust mechanism for tracking variable dependencies and verifying intermediate values, especially for weaker SLMs. We identified and summarized several possible reasons for its poor performance:

1. **Task Relevance and Reasoning Depth**: According to the original paper, Symbolic CoT is primarily designed for logical reasoning tasks (Xu et al., 2024a). However, our datasets focus on mathematical and commonsense reasoning, where the advantages of symbolic reasoning—particularly its effectiveness in handling deeper reasoning—do not manifest as clearly.

2. **Implementation Differences**: The original study employed multiple stages and corresponding special tokens to enhance symbolic reasoning. In contrast, our implementation only adopted the symbolic reasoning format without these additional mechanisms, which might have impacted its effectiveness.

3. **Pretraining Data Constraints**: SLMs have relatively limited pretraining corpora, which likely contain fewer instances of symbolic reasoning formats. As a result, weaker models struggle to acquire symbolic reasoning capabilities with only a small number of training samples.

H Student Performance across Different Teacher Models



(a) SVAMP (b) GSM8K (c) MATH

(d) AQuA-RAT (e) CommonsenseQA (f) StrategyQA

Figure 13: Student model performance across different teacher models. Each bar represents the average accuracy of a specific student model trained on CoT from different teacher models.