

TEACHER THAT LISTENS: ADAPTIVE STUDENT-AWARE DISTILLATION FOR REASONING

Anonymous authors

Paper under double-blind review

ABSTRACT

Knowledge distillation is a standard approach to compress the capabilities of large language models into smaller students. However, standard distillation methods often produce suboptimal results due to a mismatch between teacher-generated rationales and the student’s specific learning requirements. In this paper, we introduce the Adaptive student-aware Distillation for Reasoning (**AdaptDistill**), designed to bridge this gap by iteratively identifying the student’s errors and allowing the teacher to refine its explanations according to the student’s needs. Each iteration directly targets the student’s learning deficiencies, motivating the teacher to provide tailored rationales that specifically address these weaknesses for better learning. Empirical evaluations on various challenging mathematical and commonsense reasoning tasks demonstrate that our adaptive distillation approach, **AdaptDistill**, significantly outperforms standard distillation methods, achieving significant performance gains. Our work fundamentally reframes knowledge distillation as an iterative *teacher–student* interaction, effectively leveraging dynamic refinement by the teacher for better knowledge distillation.

1 INTRODUCTION

An increasingly popular approach for reducing the size and computational demands of large language models (LLMs) involves distilling their knowledge into compact, high-performing student models (Hinton et al., 2015; Sanh et al., 2020; Zhu et al., 2024). Initially, the primary approach involved aligning the student’s outputs with the soft logits of the teacher, enabling the student to approximate the teacher’s distribution (Hinton et al., 2015). However, recently, this paradigm has expanded to include reasoning chains, which seek to instill smaller models with interpretable intermediate steps rather than simply matching final outputs (Shridhar et al., 2023; Hsieh et al., 2023; Li et al., 2023; Chenglin et al., 2024; Liu et al., 2024).

While effective, these approaches still rely on a one-shot distillation process,¹ where the teacher produces rationales independently of the student’s behavior. This leads to a distributional mismatch: the teacher’s rationales often reflect its own reasoning style rather than addressing the student’s concrete mistakes or learning bottlenecks (Agarwal et al., 2024; Adarsh et al., 2025). As a result, the student may fail to fully benefit from teacher supervision, limiting both reasoning fidelity and transferability.

We propose Adaptive student-aware Distillation for Reasoning, an *iterative*, adaptive distillation framework that closes the loop between teacher and student. In each round, the teacher generates rationales to teach the given task to the student; the student learns it and attempts the task that exposes its learning gaps. The teacher then *optimizes its rationale* by conditioning on a scored history based on whether the student can solve the task correctly or not along with the learning gaps and produces a refined, *targeted* explanation. The student is then fine-tuned on a curated mix of its *own* correct traces (to preserve successful behaviors) and the teacher’s corrected ones (to address gaps). Repeating this process personalizes supervision to the student rather than reflecting generic teacher behavior.

¹We refer to training a smaller model on a larger model’s outputs as *one-shot distillation*, i.e., without any iterative student–teacher interaction.

054 This loop mimics the optimization process in the context window of the teacher as every itera-
055 tion maintains a trajectory of (rationales, score) pairs and proposes the next candidate rationale to
056 improve an *objective* of improving the student accuracy. In this sense, AdaptDistill instantiates a
057 derivative-free optimization loop driven by an LLM that reasons over its own scored attempts and the
058 student’s responses to iteratively improve guidance. An overview of the loop is shown in Figure 1.

059 We evaluate AdaptDistill on challenging mathematical reasoning benchmarks, including MATH
060 (Lightman et al., 2024) and MMLU Pro (Wang et al., 2024). Empirically, our iterative strategy con-
061 sistentlly outperforms one-shot distillation, with accuracy improvements of up to 20%. We further
062 find that the teacher-generated datasets produced by AdaptDistill retain their effectiveness across
063 diverse student architectures, and that students trained with our approach maintain (and sometimes
064 improve) out-of-domain performance. These results suggest that our framework improves not only
065 in-distribution accuracy but also the transferability of reasoning skills.

066 In summary, our work introduces a shift in perspective: distillation is no longer a one-time transfer
067 of knowledge, but an ongoing adaptive teacher–student interaction that produces more robust and
068 generalizable student models.

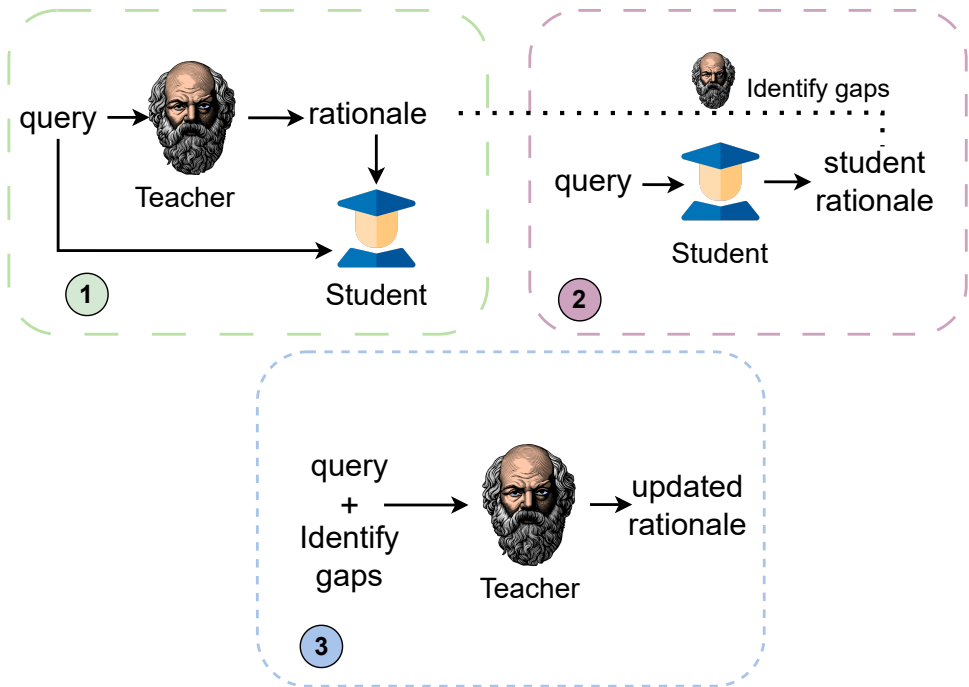
071 2 RELATED WORK

072 **Knowledge distillation** Extensive research has explored knowledge distillation since (Hinton
073 et al., 2015) introduced the concept of training smaller student models using soft-label outputs (soft
074 logits) from larger teacher models. Early influential models such as DistilBERT (Sanh et al., 2020)
075 demonstrated substantial parameter reductions (40%) without significant performance losses. Sub-
076 sequent advancements, like MiniLM (Wang et al., 2020), further refined this technique by focusing
077 specifically on distilling the self-attention mechanism. More recent works, including (Shum et al.,
078 2024) and (Timiryasov & Tastet, 2023), explored selective training strategies by choosing essential
079 samples from the teacher’s training set.

082 The integration of Chain-of-Thought (CoT) reasoning (Wei et al., 2022) into distillation has sparked
083 additional innovations. For example, (Shridhar et al., 2023) trained student models to explicitly
084 replicate the teacher’s CoT reasoning steps. Similarly, (Li et al., 2023) generated diverse ratio-
085 nales for individual questions, while (Chenglin et al., 2024) extended this by incorporating multiple
086 reasoning styles. (Zhu et al., 2024) distilled mathematical reasoning into structured equations and
087 combined various reasoning modalities (text, code, equations). Further, (Luo et al., 2025) leveraged
088 reinforcement learning to fine-tune student models. Despite these advancements, most approaches
089 remain one-directional, with teacher outputs being independent of student-specific needs. In con-
090 trast, our method explicitly addresses the student’s learning gaps by encouraging the teacher to adapt
091 their rationales based on the student needs.

092 **Iterative distillation** Our work draws significant inspiration from iterative forms of knowledge
093 distillation. Previous research often emphasized the transfer of intermediate reasoning steps to
094 smaller models, either sequentially (Hsieh et al., 2023; Magister et al., 2023; Shridhar et al., 2023)
095 or concurrently integrating various reasoning strategies (Chenglin et al., 2024; Zhu et al., 2024; Jain
096 et al., 2024). Notably, (Wang et al., 2023) employed iterative interactions between students and
097 black-box teachers, with teachers providing targeted feedback based on student failures. Similarly,
098 (Adarsh et al., 2025) combined iterative self-guided training and self-generated data to enhance rea-
099 soning in smaller models. Additionally, (Agarwal et al., 2024) utilized on-policy training to better
100 align student and teacher distributions by incorporating student-generated outputs. Self-distillation
101 methods (Zelikman et al., 2022; Liu et al., 2024) involved models generating, filtering, and fine-
102 tuning based on their own rationales. Reinforcement learning-based self-training was further ex-
103 plored by (Gulcehre et al., 2023), where off-policy sampling informed dataset generation. Unlike
104 these methods, our iterative approach uniquely emphasizes a dynamic, adaptive feedback loop where
105 the teacher explicitly targets and rectifies the student’s learning challenges.

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132



133
134
135
136
137
138
139

Figure 1: **Overview of AdaptDistill.** (1) The teacher generates an initial rationale for a query and supervises the student. (2) The student attempts the same query; the teacher compares the student’s rationale with its own prior rationale to identify *learning gaps*. (3) Re-prompted with the query plus gap information (and a scored history on the validation set), the teacher produces an *updated rationale* targeted to the student’s weaknesses. Steps 1–3 repeat: the student is fine-tuned on curated traces while the teacher keeps track of progress in its prompt, effectively acting as an in-context optimizer of rationales.

140

3 METHODOLOGY

143

144

In this section, we first outline the standard knowledge distillation process and discuss its key limitations. We then introduce our proposed Adaptive student-aware Distillation for Reasoning framework, which addresses these limitations by enabling the teacher to iteratively identify the student’s weaknesses and refine the rationale being distilled accordingly.

145

146

Problem Setup. We consider a large teacher model p_L (LLM) and a smaller student model p_{sm}^θ (with learnable parameters θ). We have a dataset of math questions $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^n$, and an associated held-out validation set $\mathcal{V} = \{(q_j, a_j)\}_{j=1}^m$ to monitor progress at each iteration. Our goal is to distill the teacher’s reasoning capabilities into the student iteratively by constantly moderating the errors made by the student and targeted refinements from the teacher. We use a fixed instruction I to elicit chain-of-thought (CoT) from both teacher and student.

147

148

149

3.1 STANDARD KNOWLEDGE DISTILLATION

150

151

We follow (Shridhar et al., 2023; Magister et al., 2023). The teacher $p_L(r | q, I)$ is prompted to produce a rationale (chain-of-thought) r_i for each question q_i . Let \hat{a}_i be the final numeric result extracted from r_i (denoted $\text{ans}(r_i)$). We keep only those samples for which $\hat{a}_i = a_i$, yielding a filtered dataset $\mathcal{D}_{LLM} = \{(q_i, r_i)\}_{i=1}^N$ of correct (question, rationale) pairs.

152

153

154

155

156

157

158

The student is then fine-tuned by maximizing the log-likelihood of these rationales:

$$\mathcal{L}_L(\theta) = -\frac{1}{|\mathcal{D}_{\text{LLM}}|} \sum_{(q_i, r_i) \in \mathcal{D}_{\text{LLM}}} \sum_{t=1}^{M_i} \log p_{\text{sm}}^\theta(r_{i,t} | r_{i,<t}, q_i, I), \quad (1)$$

where M_i is the number of tokens in r_i . Minimizing $\mathcal{L}_L(\theta)$ via gradient descent adjusts θ so that the student mimics the teacher’s rationales. This is the baseline *one-step* standard knowledge distillation.

3.2 ADAPTIVE STUDENT-AWARE DISTILLATION FOR REASONING

Standard Knowledge Distillation can result in a discrepancy between the output distributions of the teacher and student models, which may hinder effective learning (Agarwal et al., 2024; Adarsh et al., 2025). To counter this, we propose an *iterative* distillation strategy `AdaptDistill` that repeatedly identifies the student’s errors and lets the teacher produce improved rationales that focus on the *learning gaps* to improve the student’s performance. The setup is visualized in Figure 1.

We perform K such iterations, maintaining updated student parameters $\theta^{(k)}$ after each round. Each step is explained below.

1) Identify Student Learning Gaps & Incorporate in Teacher Prompt. At each iteration $k \in \{1, \dots, K\}$, the student $p_{\text{sm}}^{\theta^{(k-1)}}(r | q, I)$ answers each problem in the validation set \mathcal{V} . The student’s generation for $q_j \in \mathcal{V}$ is denoted $\tilde{r}_j^{(k, \text{student})}$. We compare the final numeric result $\text{ans}(\tilde{r}_j^{(k, \text{student})})$ to a_j and obtain a binary score $\text{score}_j^{(k)} \in \{0, 1\}$. We collect these labeled pairs and maintain a running history across iterations:

$$\mathcal{H}^{(k)} := \bigcup_{\ell=1}^k \left\{ (q_j, \tilde{r}_j^{(\ell, \text{student})}, \text{score}_j^{(\ell)}) \right\}_{j=1}^m.$$

This history is shown to the teacher in its prompt context so the teacher can observe the student’s progress and mistakes.

2) Regenerate from the Teacher to adapt to the student needs. We now consider the *training* dataset \mathcal{D} to improve the student. For each $(q_i, a_i) \in \mathcal{D}$, retrieve: (i) the student’s most recent generation for q_i ,

$$\tilde{r}_i^{(\leq k-1, \text{student})} := \begin{cases} \tilde{r}_i^{(k-1, \text{student})}, & \text{if available,} \\ \tilde{r}_i^{(\text{init})}, & \text{otherwise,} \end{cases}$$

and (ii) the teacher’s previous rationale $r_i^{(k-1, \text{teacher})}$ (if it exists). Define the indicator of the student being already correct:

$$S_i^{(k-1)} := \mathbf{1} \left\{ \text{ans}(\tilde{r}_i^{(\leq k-1, \text{student})}) = a_i \right\}.$$

Use-student rule. If $S_i^{(k-1)} = 1$ (the student is already correct), we *do not* query the teacher; instead, we add the student’s own rationale $\tilde{r}_i^{(\leq k-1, \text{student})}$ to the training set for this question. This was done following (Adarsh et al., 2025) and (Agarwal et al., 2024) as adding student’s own generations helps in reducing the distributional mismatch between the teacher and the student.

Otherwise (i.e., if $S_i^{(k-1)} = 0$), we construct gap information

$$\Delta_i^{(k)} := (\tilde{r}_i^{(\leq k-1, \text{student})}, r_i^{(k-1, \text{teacher})}, \mathcal{H}^{(k)}),$$

and sample a new teacher rationale

$$r_i^{(k, \text{teacher})} \sim p_L(\cdot | q_i, I, \Delta_i^{(k)}).$$

Let $C_i^{(k)} := \mathbf{1} \{ \text{ans}(r_i^{(k, \text{teacher})}) = a_i \}$ denote the acceptance event.

The curated dataset for iteration k then aggregates correct student traces and (filtered) teacher traces:

$$\mathcal{D}_{\text{LLM}}^{(k)} := \underbrace{\left\{ (q_i, \tilde{r}_i^{(\leq k-1, \text{student})}) : S_i^{(k-1)} = 1 \right\}}_{\text{student already correct}} \cup \underbrace{\left\{ (q_i, r_i^{(k, \text{teacher})}) : S_i^{(k-1)} = 0, C_i^{(k)} = 1 \right\}}_{\text{regenerate with teacher and filter by correctness}}.$$

Algorithm 1 Adaptive student-aware Distillation for Reasoning

Require: Teacher p_L ; **Student** p_{sm}^θ ; **Train set** $\mathcal{D}=\{(q_i, a_i)\}_{i=1}^n$; **Val set** $\mathcal{V}=\{(q_j, a_j)\}_{j=1}^m$; **Instruction** I ; **Max iters** K .

- 1: Initialize $\theta^{(0)}$ (e.g., from one-step distillation); set history $\mathcal{H} \leftarrow \emptyset$.
- 2: **for** $k = 1$ **to** K **do**
- 3: **// Step 1: Identify learning gaps**
- 4: **for each** (q, a) **in** \mathcal{V} **do**
- 5: $r_s \leftarrow \text{GENERATE}(p_{\text{sm}}^{\theta^{(k-1)}}, q, I)$
- 6: $score \leftarrow \text{CORRECT}(r_s, a)$ \triangleright 1 if correct, 0 otherwise
- 7: $\mathcal{H} \leftarrow \mathcal{H} \cup \{(q, r_s, score)\}$ \triangleright optionally cap to recent h items
- 8: **end for**
- 9: **// Step 2: Build curated training set (use-student or regenerate)**
- 10: $\mathcal{D}_{\text{LLM}}^{(k)} \leftarrow \emptyset$
- 11: **for each** (q, a) **in** \mathcal{D} **do**
- 12: $r_s \leftarrow \text{MOSTRECENTSTUDENT}(q)$ \triangleright cached from $k-1$, else $\text{GENERATE}(\cdot)$ once
- 13: **if** $\text{CORRECT}(r_s, a)$ **then** \triangleright student already correct
- 14: $\mathcal{D}_{\text{LLM}}^{(k)} \leftarrow \mathcal{D}_{\text{LLM}}^{(k)} \cup \{(q, r_s)\}$ \triangleright use-student rule
- 15: **else**
- 16: $\Delta \leftarrow \text{BUILDGAPINFO}(q, r_s, \text{PREVTEACHER}(q), \mathcal{H})$
- 17: $r_t \leftarrow \text{GENERATE}(p_L, q, I, \Delta)$
- 18: **if** $\text{CORRECT}(r_t, a)$ **then** \triangleright accept only if final answer matches
- 19: $\mathcal{D}_{\text{LLM}}^{(k)} \leftarrow \mathcal{D}_{\text{LLM}}^{(k)} \cup \{(q, r_t)\}$
- 20: **end if**
- 21: **end if**
- 22: **end for**
- 23: **// Step 3: Student update**
- 24: $\theta^{(k)} \leftarrow \text{TRAINSTUDENT}(\theta^{(k-1)}, \mathcal{D}_{\text{LLM}}^{(k)}, I)$ \triangleright standard teacher-forced LM training on curated rationales
- 25: **// Step 4: Convergence check**
- 26: $val_acc \leftarrow \text{EVAL}(p_{\text{sm}}^{\theta^{(k)}}, \mathcal{V}, I)$
- 27: **if** $\text{NOIMPROVEMENT}(val_acc)$ **then**
- 28: **break**
- 29: **end if**
- 30: **end for**

3) Fine-tune the Student. We update the student parameters by minimizing the negative log-likelihood of the teacher’s new rationales:

$$\mathcal{L}_L^{(k)}(\theta) = -\frac{1}{|\mathcal{D}_{\text{LLM}}^{(k)}|} \sum_{(q_i, r_i^{(k, \text{teacher})}) \in \mathcal{D}_{\text{LLM}}^{(k)}} \sum_{t=1}^{M_i^{(k)}} \log p_{\text{sm}}^\theta(r_{i,t}^{(k, \text{teacher})} | r_{i, < t}^{(k, \text{teacher})}, q_i, I), \quad (2)$$

where $M_i^{(k)}$ is the token length of $r_i^{(k, \text{teacher})}$. The updated parameters are

$$\theta^{(k)} = \arg \min_{\theta} \mathcal{L}_L^{(k)}(\theta). \quad (3)$$

4) Convergence Check. We repeat Steps 1–3 until no further improvement is observed on the validation set or until we reach a maximum iteration K .

Algorithm 1 summarizes the proposed adaptive distillation strategy in pseudo code.

4 EXPERIMENTAL DETAILS

4.1 DATASET

We use the Numina Math dataset (LI et al., 2024) to fine-tune the student models, as each solution is formatted in a Chain-of-Thought (CoT) style. We filter to math *word problems* and further preprocess the corpus to remove samples whose answers are “proof”, “notfound”, empty, or whose problem/solution fields are invalid. Because Numina Math does not include a validation set, we partition the training set and randomly sample $|\mathcal{V}| = 20$ examples to create a held-out validation set. Since the validation history is kept in the teacher’s prompt and grows linearly with iterations, we choose a small validation set so the prompt remains within the teacher’s context window even after K iterations.

For testing, we aggregate standard mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), MMLU Pro (Wang et al., 2024), MATH 500 (Lightman et al., 2024), and SVAMP (Patel et al., 2021). The overall distribution is shown in Table 1. Additionally, we evaluate out-of-domain (OOD) generalization on StrategyQA (Geva et al., 2021) (687 test samples²) and TheoremQA (Chen et al., 2023) (800 test samples), using 4-shot CoT for StrategyQA and 5-shot CoT for TheoremQA.

Table 1: Distribution of training and test datasets.

Dataset	Training Set (N = 550,000)		Test Set (N = 4170)	
	# Samples	Dist. (%)	# Samples	Dist. (%)
NuminaMath (LI et al., 2024)	550,000	100%	-	-
MATH 500 (Lightman et al., 2024)	-	-	500	12%
GSM8K (Cobbe et al., 2021)	-	-	1319	32%
MMLU Pro (Wang et al., 2024)	-	-	1351	32%
SVAMP (Patel et al., 2021)	-	-	1000	24%

4.2 MODELS

We employ llama-3.2-70B (Dubey et al., 2024) as the teacher model and consider three student models—llama-3.3-1B, qwen-2.5-1.5B (Bai et al., 2023), and SmoLLM2-1.7B (Al-lal et al., 2025). The student models are chosen for their diversity in size and baseline performance. Across experiments, the iterative procedure saturates by $K=4$, with optimal results observed at $K=3$. For each student, we train 5 epochs at iteration $K=1$ and 3 epochs at subsequent iterations. Training each student requires ~ 10 GPU hours per epoch on H100 GPUs. Student inference over the training set consumes ~ 240 GPU hours on GH200 GPUs, and teacher data generation takes ~ 800 GPU hours per iteration on GH200s. Decoding uses vLLM (Kwon et al., 2023) with temperature 0.7.

Metrics. We report $\text{acc}@1$ after numeric answer normalization; “Average” denotes the *micro-average*, i.e., accuracy weighted by per-dataset test counts in Table 1.

5 RESULTS AND DISCUSSION

Iterative adaptive distillation improves student performance. Table 2 presents the results of three different student models: Qwen2.5 1.5B, SmoLLM2 1.7B, and Llama-3.2 1B over three iterations (K) of our iterative distillation procedure AdaptDistill. We report performance on four

²Test samples taken from <https://huggingface.co/datasets/ChilleD/StrategyQA>

Table 2: acc@1 for three student models over iterations (Iter.). Gains (\uparrow/\downarrow) are relative to Iter 1 of the same model (standard distillation). Underlining indicates statistical significant results with $p < 0.05$ vs. baseline using a paired test on item-level predictions. “Average” is the micro-average weighted by per-dataset test counts in Table 1.

Model	Iter.	GSM8K	MATH	MMLU PRO	SVAMP	Average
Qwen	1	50.95	32.80	14.51	76.70	43.14
	2	<u>53.22</u> ($\uparrow+2.27$)	<u>35.80</u> ($\uparrow+3.00$)	15.40 ($\uparrow+0.89$)	<u>84.20</u> ($\uparrow+7.50$)	<u>46.31</u> ($\uparrow+3.17$)
	3	<u>55.04</u> ($\uparrow+4.09$)	<u>39.00</u> ($\uparrow+6.20$)	15.17 ($\uparrow+0.66$)	<u>87.40</u> ($\uparrow+10.70$)	<u>47.96</u> ($\uparrow+4.82$)
SmolLM	1	53.60	30.60	13.47	85.50	45.49
	2	<u>55.72</u> ($\uparrow+2.12$)	<u>34.20</u> ($\uparrow+3.60$)	<u>15.03</u> ($\uparrow+1.56$)	<u>87.20</u> ($\uparrow+1.70$)	47.51 ($\uparrow+2.02$)
	3	<u>56.93</u> ($\uparrow+3.33$)	<u>38.00</u> ($\uparrow+7.40$)	<u>16.21</u> ($\uparrow+2.74$)	<u>87.20</u> ($\uparrow+1.70$)	<u>48.70</u> ($\uparrow+3.21$)
Llama	1	34.72	20.60	9.74	79.50	35.59
	2	<u>39.34</u> ($\uparrow+4.62$)	<u>24.80</u> ($\uparrow+4.20$)	<u>12.40</u> ($\uparrow+2.66$)	<u>78.50</u> ($\downarrow-1.00$)	<u>38.27</u> ($\uparrow+2.68$)
	3	<u>43.96</u> ($\uparrow+9.24$)	<u>27.00</u> ($\uparrow+6.40$)	<u>15.06</u> ($\uparrow+5.32$)	<u>80.50</u> ($\uparrow+1.00$)	<u>40.95</u> ($\uparrow+5.36$)

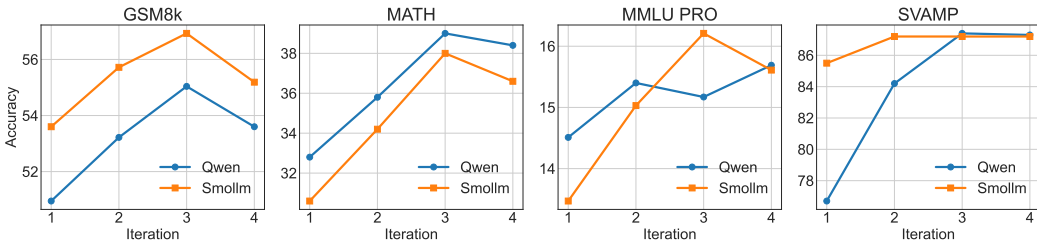


Figure 2: Accuracy (acc@1) for Qwen and SmolLM2 models on GSM8K, MATH, MMLU Pro and SVAMP datasets for 4 iterations.

mathematical reasoning datasets: GSM8K, MATH, MMLU PRO, and SVAMP, along with an Average score across these tasks. We observe a consistent improvement over all datasets across all models. For instance, under Qwen2.5 1.5B, the performance on GSM8K increases from 50.95 at iteration 1 to 55.04 at iteration 3. A similar trend is observed for the MATH dataset, which improves from 32.80 \rightarrow 39.00, and on SVAMP which increases from 76.70 \rightarrow 87.40. SmolLM2 and Llama-3.2 also follow the same upward trajectory. Notably, SmolLM2 gains +7.40 points on MATH by iteration 3 (from 30.60 \rightarrow 38.00) and +2.74 points on MMLU PRO (from 13.47 \rightarrow 16.21). Llama-3.2 sees gains of +9.24 and +8.40 points on GSM8K and MATH, respectively, moving from 34.72 \rightarrow 43.96 on GSM8K and 20.60 \rightarrow 29.00 on MATH. Overall, each model consistently benefits from iterative distillation, achieving higher accuracy on each dataset from iteration 1 to iteration 3.

Deciding when to stop. Figure 2 visualizes Qwen2.5 and SmolLM2 across four iterations on the same four benchmarks, mirroring the upward trend through Iter 3 in Table 2. At Iter 4, accuracy slightly declines (e.g., Qwen2.5: 55.04 \rightarrow 53.60 on GSM8K; SmolLM2: 56.93 \rightarrow 55.19 on GSM8K), indicating convergence. Since these iteration 4 numbers are lower than iteration 3 results, we conclude that the model has converged and additional regeneration from the teacher is no longer beneficial, prompting us to stop further distillation steps at iteration 3.

378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

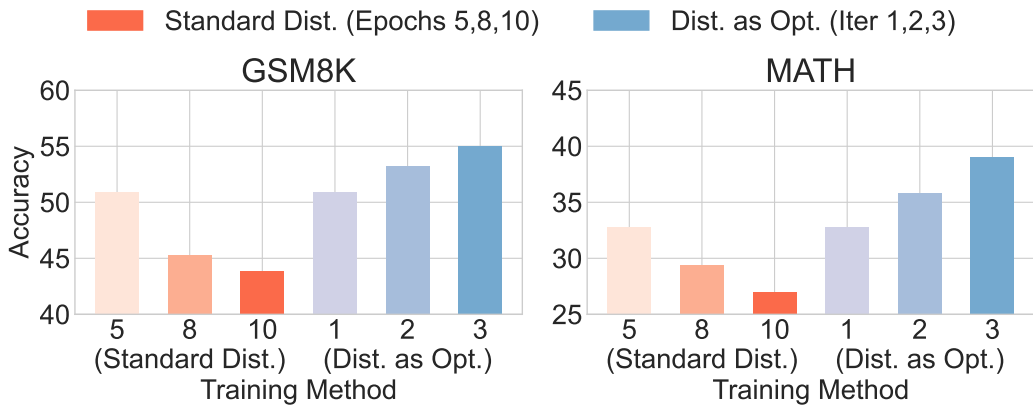


Figure 3: Accuracy comparison of training standard distillation and AdaptDistill for the equal number of epochs. Note that iteration 1 for AdaptDistill is same as epoch 5 for standard distillation.

Table 3: Comparison of acc@1 accuracy for Qwen and SmoLLM across four benchmarks (GSM8K, MATH, MMLU PRO, SVAMP) and the resulting average score. Each model can be trained on data generated by its own teacher or a cross-teacher (e.g., Qwen data for SmoLLM2). Gains (\uparrow or \downarrow) are shown relative to the baseline (standard distillation). Underline indicates statistical significance with $p < 0.05$ compared to the baseline.

	Teacher	GSM8K	MATH	MMLU PRO	SVAMP	Average
Qwen	Baseline	50.95	32.80	14.51	76.70	43.14
	Qwen	<u>56.10</u> ($\uparrow+5.15$)	<u>39.40</u> ($\uparrow+6.60$)	15.98 ($\uparrow+1.47$)	<u>86.50</u> ($\uparrow+9.80$)	<u>48.39</u> ($\uparrow+5.52$)
	SmoLLM2	<u>54.00</u> ($\uparrow+3.05$)	<u>34.00</u> ($\uparrow+1.20$)	14.35 ($\downarrow-0.16$)	<u>86.50</u> ($\uparrow+9.80$)	<u>46.57</u> ($\uparrow+3.43$)
SmoLLM	Baseline	53.60	30.60	13.47	85.50	45.49
	Qwen	<u>56.86</u> ($\uparrow+3.26$)	<u>32.80</u> ($\uparrow+2.20$)	14.95 ($\uparrow+1.48$)	85.60 ($\uparrow+0.10$)	47.29 ($\uparrow+1.80$)
	SmoLLM2	<u>56.86</u> ($\uparrow+3.26$)	<u>33.60</u> ($\uparrow+3.00$)	<u>15.00</u> ($\uparrow+1.53$)	86.80 ($\uparrow+1.30$)	<u>47.72</u> ($\uparrow+2.23$)

Can standard distillation trained for same number of epochs perform similar to AdaptDistill? Figure 3 compares standard distillation (continued epochs) to AdaptDistill under the same total number of epochs. Extending standard training (e.g., 10 epochs) yields diminished returns and overfitting (GSM8K 43.90%, MATH 27.00%, MMLU Pro 13.82%, SVAMP 72.90%, average 39.41%), all below the Iter 1 baseline. In contrast, AdaptDistill improves each round: for Qwen2.5, GSM8K 50.95% \rightarrow 55.04%, MATH 32.80% \rightarrow 39.00%, MMLU Pro 14.51% \rightarrow 15.17%, SVAMP 76.70% \rightarrow 87.40%, raising the average from 43.14% to 47.96%. Iterative, gap-aware regeneration is therefore more effective than simply training longer.

Does AdaptDistill create a better teacher-generated dataset? Table 3 examines training each student for 5 epochs on datasets generated at the *final* iteration of the teacher. The column “Teacher” denotes which student’s gap signals were used to condition the (fixed) llama-3.2-70B teacher during data generation (“Baseline” is standard distillation). For Qwen2.5, using Qwen-conditioned teacher generations yields sizable gains (GSM8K 50.95 \rightarrow 56.10, SVAMP 76.70 \rightarrow 86.50, average +5.52). Even SmoLLM2-conditioned teacher data improves Qwen over baseline (average +3.43). SmoLLM2 exhibits the same pattern: with SmoLLM2-conditioned teacher data, GSM8K rises to 56.86 and MMLU Pro to 15.00, while Qwen-conditioned data also helps (average +1.80). These

Table 4: Performance (%) on **StrategyQA** and **Theorem QA** benchmarks for Qwen and SmoLLM2 models. Standard Distillation represents the results from Iteration 1 while **AdaptDistill** represents the results from Iteration 3 from [Table 2](#). Gains (\uparrow) are shown in parentheses.

Model	StrategyQA		Theorem QA	
	Standard Dist.	AdaptDistill	Standard Dist.	AdaptDistill
Qwen	57.50%	60.00% ($\uparrow+2.50$)	7.5%	8.125% ($\uparrow+0.625$)
SmoLLM2	60.20%	63.00% ($\uparrow+2.80$)	5.62%	6.25% ($\uparrow+0.63$)

results suggest the iterative procedure produces teacher outputs that are *personalized* to a student’s weaknesses and transfer partly across students.

Does AdaptDistill cause overfitting to mathematical task? To assess overfitting, we evaluate on out of domain (OOD) datasets: StrategyQA and TheoremQA. [Table 4](#) shows Qwen and SmoLLM2 both improve from Iter 1 (standard distillation) to Iter 3 (**AdaptDistill**) on **both** OOD tasks. For Qwen: +2.50% on StrategyQA and +0.625% on TheoremQA; for SmoLLM2: +2.80% and +0.63%, respectively. These gains suggest that **AdaptDistill** enhances general reasoning rather than overfitting to mathematical problems. We hypothesize that mathematical reasoning serves as a structured training signal. The explicit CoT traces learned in math tasks generalize to tasks requiring multi-hop reasoning and factual consistency, as observed on StrategyQA. Our aim here is not to maximize absolute OOD scores, but to test whether the adaptive procedure compromises out-of-domain performance; empirically, it does not.

6 CONCLUSION

We introduced an iterative distillation paradigm **AdaptDistill** aimed at bridging the performance gap between large teacher models and smaller student models more effectively than conventional one-shot distillation. By repeatedly identifying the student’s errors, regenerating teacher rationales that address these *learning gaps*, and re-training the student on this tailored data, our method yields consistent improvements on challenging mathematical benchmarks. Experimental results indicate that **AdaptDistill** not only avoids overfitting often seen in prolonged standard distillation, but also makes the teacher more attuned to student-specific weaknesses, effectively “personalizing” the teacher’s rationales. Empirically, we observed that final teacher-generated data after multiple rounds of iteration significantly enhances student performance, with gains of up to 20% compared to the baseline. Even cross-teacher data (e.g., teacher data for one student used on another) shows improved results over standard approaches. These findings underscore that incorporating iterative feedback loops between the teacher and the student leads to a more specialized and effective transfer of reasoning knowledge.

7 ETHICS STATEMENT

This work uses math problems from publicly available datasets such as NuminaMath, GSM8K, MATH, SVAMP and MMLU Pro for fine-tuning and evaluation. These datasets are widely used in prior research and do not involve sensitive or personally identifiable information. The validation set for our prompts was also created from the same sources.

We acknowledge that large language models may produce harmful or unsafe outputs; developing safeguards to mitigate such risks remains important future work. All references have been properly cited, and we have adhered to standard research integrity and ethical practices throughout this project.

8 REPRODUCIBILITY STATEMENT

We have taken several measures to ensure reproducibility:

- **Datasets:** We use only publicly available datasets (NuminaMath, GSM8K, MATH, MMLU Pro, SVAMP), with preprocessing and prompts described in section 4.1 and appendix A.
- **Models:** The teacher and student model architectures are explicitly stated (LLaMA-3.2-70B: **teacher**; Qwen-2.5-1.5B, SmolLM2-1.7B, LLaMA-3.3-1B: **students**).
- **Training Setup:** Hyperparameters, iteration schedules, and optimization details are fully specified in section 4.2.
- **Compute Resources:** We report GPU hours and hardware details used for both student training and teacher generation in section 4.2.
- **Code Availability:** All source code, data preprocessing scripts, and evaluation pipelines will be released upon publication to facilitate exact reproduction.

REFERENCES

- Shivam Adarsh, Kumar Shridhar, Caglar Gulcehre, Nicholas Monath, and Mrinmaya Sachan. SIKeD: Self-guided iterative knowledge distillation for mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9868–9880. Association for Computational Linguistics, July 2025. doi: 10.18653/v1/2025.findings-acl.513. URL <https://aclanthology.org/2025.findings-acl.513/>.
- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=3zKtaqxLhW>.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. SmolLM2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025. URL <https://arxiv.org/abs/2502.02737>.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint*, 2023. URL <https://arxiv.org/abs/2309.16609>.
- Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.489. URL <https://aclanthology.org/2023.emnlp-main.489/>.
- Li Chenglin, Qianglong Chen, Liangyue Li, Caiyu Wang, Feng Tao, Yicheng Li, Zulong Chen, and Yin Zhang. Mixed distillation helps smaller language models reason better. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1673–1690, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.91. URL <https://aclanthology.org/2024.findings-emnlp.91/>.
- 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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint*, 2024. URL <https://arxiv.org/abs/2407.21783>.

- 540 Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle
541 use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions*
542 *of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl.a.00370.
543 URL <https://aclanthology.org/2021.tacl-1.21/>.
- 544
545 Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek
546 Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud
547 Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling,
548 2023. URL <https://arxiv.org/abs/2308.08998>.
- 549 Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.
550 URL <https://arxiv.org/abs/1503.02531>.
- 551
552 Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner,
553 Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger
554 language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-
555 Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics:*
556 *ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguis-
557 tics. doi: 10.18653/v1/2023.findings-acl.507. URL [https://aclanthology.org/2023.
558 findings-acl.507/](https://aclanthology.org/2023.findings-acl.507/).
- 559 Kushal Jain, Moritz Miller, Niket Tandon, and Kumar Shridhar. First-step advantage: Importance of
560 starting right in multi-step math reasoning, 2024. URL [https://arxiv.org/abs/2311.
561 07945](https://arxiv.org/abs/2311.07945).
- 562
563 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
564 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
565 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operat-*
566 *ing Systems Principles*, 2023. URL [https://dl.acm.org/doi/10.1145/3600006.
567 3613165](https://dl.acm.org/doi/10.1145/3600006.3613165).
- 568
569 Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang,
570 Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann
571 Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. [[https://huggingface.
572 co/AI-MO/NuminaMath-1.5](https://huggingface.co/AI-MO/NuminaMath-1.5)] ([https://github.com/project-numina/
573 aimo-progress-prize/blob/main/report/numina_dataset.pdf](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)), 2024.
- 574
575 Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic
576 chain-of-thought distillation: Small models can also “think” step-by-step. In Anna Rogers, Jordan
577 Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Associ-*
578 *ation for Computational Linguistics (Volume 1: Long Papers)*, pp. 2665–2679, Toronto, Canada,
579 July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.150. URL
580 <https://aclanthology.org/2023.acl-long.150/>.
- 581
582 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
583 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth*
584 *International Conference on Learning Representations*, 2024. URL [https://openreview.
585 net/forum?id=v8L0pN6EOi](https://openreview.net/forum?id=v8L0pN6EOi).
- 586
587 Rongxing Liu, Kumar Shridhar, Manish Prajapat, Patrick Xia, and Mrinmaya Sachan. Smart: Self-
588 learning meta-strategy agent for reasoning tasks. *arXiv preprint arXiv:2410.16128*, 2024. URL
589 <https://arxiv.org/abs/2410.16128>.
- 590
591 Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng,
592 Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering
593 mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth*
594 *International Conference on Learning Representations*, 2025. URL [https://openreview.
595 net/forum?id=mMPMHWOdOy](https://openreview.net/forum?id=mMPMHWOdOy).
- 596
597 Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn.
598 Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the*

- 594 *Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781. Association
595 for Computational Linguistics, July 2023. URL [https://aclanthology.org/2023.](https://aclanthology.org/2023.acl-short.151)
596 [acl-short.151](https://aclanthology.org/2023.acl-short.151).
597
- 598 Arkil Patel, S. Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math
599 word problems? In *North American Chapter of the Association for Computational Linguistics*,
600 2021. URL <https://api.semanticscholar.org/CorpusID:232223322>.
601
- 602 Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version
603 of bert: smaller, faster, cheaper and lighter, 2020. URL [https://arxiv.org/abs/1910.](https://arxiv.org/abs/1910.01108)
604 [01108](https://arxiv.org/abs/1910.01108).
605
- 606 Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities
607 into smaller language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki
608 (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7059–7073,
609 Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.
610 [findings-acl.441](https://aclanthology.org/2023.findings-acl.441). URL <https://aclanthology.org/2023.findings-acl.441>.
611
- 612 KaShun Shum, Minrui Xu, Jianshu Zhang, Zixin Chen, Shizhe Diao, Hanze Dong, Jipeng Zhang,
613 and Muhammad Omer Raza. FIRST: Teach a reliable large language model through efficient
614 trustworthy distillation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Pro-*
615 *ceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp.
616 12646–12659, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
617 doi: 10.18653/v1/2024.emnlp-main.703. URL [https://aclanthology.org/2024.](https://aclanthology.org/2024.emnlp-main.703/)
[emnlp-main.703/](https://aclanthology.org/2024.emnlp-main.703/).
618
- 619 Inar Timiryasov and Jean-Loup Tastet. Baby llama: knowledge distillation from an ensemble of
620 teachers trained on a small dataset with no performance penalty. In Alex Warstadt, Aaron Mueller,
621 Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjape,
622 Adina Williams, Tal Linzen, and Ryan Cotterell (eds.), *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pp. 279–289, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.
623 [conll-babyllm.24](https://aclanthology.org/2023.conll-babyllm.24/). URL <https://aclanthology.org/2023.conll-babyllm.24/>.
624
625
- 626 Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm:
627 Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In
628 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788. Curran Associates, Inc.,
629 2020. URL [https://proceedings.neurips.cc/paper_files/paper/2020/](https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
630 [file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
631
- 632 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming
633 Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang,
634 Rongqi Fan, Xiang Yue, and Wenhui Chen. MMLU-pro: A more robust and challenging multi-
635 task language understanding benchmark. In *The Thirty-eight Conference on Neural Information*
636 *Processing Systems Datasets and Benchmarks Track*, 2024. URL [https://openreview.](https://openreview.net/forum?id=y10DM6R2r3)
637 [net/forum?id=y10DM6R2r3](https://openreview.net/forum?id=y10DM6R2r3).
638
- 639 Zhaoyang Wang, Shaohan Huang, Yuxuan Liu, Jiahai Wang, Minghui Song, Zihan Zhang, Haizhen
640 Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Democratizing reasoning ability:
641 Tailored learning from large language model. In *The 2023 Conference on Empirical Methods*
642 *in Natural Language Processing*, 2023. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=rq4UfmpRA9)
643 [rq4UfmpRA9](https://openreview.net/forum?id=rq4UfmpRA9).
644
- 645 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi,
646 Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language
647 models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL [https://openreview.net/](https://openreview.net/forum?id=_VjQlMeSB_J)
[forum?id=_VjQlMeSB_J](https://openreview.net/forum?id=_VjQlMeSB_J).

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_3ELRdg2sgI.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. Distilling mathematical reasoning capabilities into small language models. *Neural Netw.*, 179(C), November 2024. ISSN 0893-6080. doi: 10.1016/j.neunet.2024.106594. URL <https://doi.org/10.1016/j.neunet.2024.106594>.

A APPENDIX

Prompt file given to the teacher in the first iteration.

You are an AI tutor tasked with improving a student's understanding of mathematical problem-solving. You will be given a question, a teacher's answer, a student's answer, and a score. Your job is to analyze these inputs and create a new answer that will help the student learn better.

Here are some examples of the task:

question:

Jim spends 8 hours scuba diving. In that time he finds a treasure chest with 100 gold coins in it. He also finds some smaller bags that have half as much gold each. He finds 25 gold coins per hour. How many smaller bags did he find?

teacher answer:

Jim finds 25 gold coins per hour, and he spends 8 hours scuba diving, so he finds a total of $25 * 8 = 200$ gold coins.

He finds a treasure chest with 100 gold coins in it, so the remaining gold coins he finds are $200 - 100 = 100$ gold coins.

Since the smaller bags have half as much gold as the treasure chest, each smaller bag has $100 / 2 = 50$ gold coins.

Therefore, the number of smaller bags he finds is $100 / 50 = \boxed{2}$ smaller bags.

Final Answer: 2

student answer:

To determine how many smaller bags Jim found, we need to follow these steps:

1. **Calculate the total amount of gold coins Jim finds over the 8 hours:** Jim finds 25 gold coins per hour for 8 hours. $[25 \text{ \textit{coins/hour}}] \times 8 \text{ \textit{ hours}} = 200 \text{ \textit{ coins}} \backslash$
2. **Determine the amount of gold coins in the smaller bags:** Let (x) be the number of smaller bags. Each smaller bag has half as much gold as a treasure chest. Since a treasure chest has 100 gold coins, each smaller bag has: $[\frac{100 \text{ \textit{ coins}}}{2} = 50 \text{ \textit{ coins}} \backslash$ Therefore, the total amount of gold coins in the smaller bags is: $[50x \text{ \textit{ coins}} \backslash$
3. **Set up the equation for the total gold coins found:** The total gold coins found by Jim is the sum of the gold coins in the treasure chest and the gold coins in the smaller bags. According to the problem, this total is 200 coins. $[100 \text{ \textit{ coins}} + 50x \text{ \textit{ coins}} = 200 \text{ \textit{ coins}} \backslash$

702 4. **Solve for x :** Subtract 100 coins from both sides of the
703 equation: $50x = 100$ Divide both sides by 50: $x = \frac{100}{50}$
704 $= 2$ Thus, Jim found $\boxed{2}$ smaller bags.
705 Final Answer: 2
706 ### score:
707 1
708
709 -----similar such examples from the validation set-----
710
711
712 First, carefully analyze the student's answer. Compare it to the
713 teacher's answer and identify any mistakes or areas where the student's
714 reasoning could be improved. Consider the following:
715
716 1. Did the student understand the problem correctly?
717 2. Did they use the right approach to solve the problem?
718 3. Are there any calculation errors?
719 4. Is their reasoning clear and logical?
720 5. Did they miss any important steps?
721
722 Next, craft a new answer that addresses the student's misunderstandings
723 or reinforces correct thinking. Your new answer should:
724
725 1. Use clear, step-by-step reasoning
726 2. Explain any concepts the student may have misunderstood
727 3. Provide additional context or examples if necessary
728 6. Use the same calculation format as the teacher's answer. If the
729 teacher answer involves latex and involves terms like $\frac{1}{2}$, π make
730 sure to extract the answer with the necessary latex keywords.
731 5. All the teacher's final answers end inside " $\boxed{\quad}$ ". Ensure, that
732 your answers also follow this format.
733 6. Lead to the correct final answer
734
735 Write your new answer using the following format:
736
737 ### new_answer
738 [Step-by-step reasoning with calculations in the format shown above]
739 Final Answer: [Correct numerical/latex answer]
740
741
742 Remember, your goal is to help the student learn and improve their
743 problem-solving skills. Focus on explaining the reasoning clearly and
744 addressing any specific issues in the student's original answer.
745
746
747

Prompt file given to the teacher in the second iteration.

748
749
750 You are an AI tutor tasked with improving a student's understanding of
751 mathematical problem-solving. You will be given a question, a teacher's
752 answer, a student's answer, and a score. Your job is to analyze these
753 inputs and create a new answer that will help the student learn better.
754
755 Here are some examples of the task:

756 ### question:
757 Jim spends 8 hours scuba diving. In that time he finds a treasure chest
758 with 100 gold coins in it. He also finds some smaller bags that have half
759 as much gold each. He finds 25 gold coins per hour. How many smaller bags
760 did he find?
761 ### ITERATION 1:
762 ### teacher answer:
763 Jim finds 25 gold coins per hour, and he spends 8 hours scuba diving, so
764 he finds a total of $25 * 8 = 200$ gold coins.
765 He finds a treasure chest with 100 gold coins in it, so the remaining
766 gold coins he finds are $200 - 100 = 100$ gold coins.
767 Since the smaller bags have half as much gold as the treasure chest, each
768 smaller bag has $100 / 2 = 50$ gold coins.
769 Therefore, the number of smaller bags he finds is $100 / 50 = \boxed{2}$
770 smaller bags.
771 Final Answer: 2
772 ### student answer:
773 To determine how many smaller bags Jim found, we need to follow these
774 steps:
775 1. **Calculate the total amount of gold coins Jim finds over the 8
776 hours:** Jim finds 25 gold coins per hour for 8 hours. $[25 \text{ \textit{coins/hour}} \times 8 \text{ \textit{ hours}} = 200 \text{ \textit{ coins}}]$
777 $[25 \text{ \textit{coins/hour}} \times 8 \text{ \textit{ hours}} = 200 \text{ \textit{ coins}}]$
778 2. **Determine the amount of gold coins in the smaller bags:** Let $(x$
779 $)$ be the number of smaller bags. Each smaller bag has half as much gold
780 as a treasure chest. Since a treasure chest has 100 gold coins, each
781 smaller bag has: $[\frac{100 \text{ \textit{ coins}}}{2} = 50 \text{ \textit{ coins}}]$
782 $[\frac{100 \text{ \textit{ coins}}}{2} = 50 \text{ \textit{ coins}}]$
783 Therefore, the total amount of gold coins in the smaller bags is: $[50x$
784 $\text{ \textit{ coins}}]$
785 3. **Set up the equation for the total gold coins found:** The total gold
786 coins found by Jim is the sum of the gold coins in the treasure chest and
787 the gold coins in the smaller bags. According to the problem, this total
788 is 200 coins. $[100 \text{ \textit{ coins}} + 50x \text{ \textit{ coins}} = 200 \text{ \textit{ coins}}]$
789 $[100 \text{ \textit{ coins}} + 50x \text{ \textit{ coins}} = 200 \text{ \textit{ coins}}]$
790 4. **Solve for (x) :** Subtract 100 coins from both sides of the
791 equation: $[50x = 100]$ Divide both sides by 50: $[x = \frac{100}{50}$
792 $= 2]$ Thus, Jim found $(\boxed{2})$ smaller bags.
793 Final Answer: 2
794 ### score:
795 1
796 ### ITERATION 2:
797 ### teacher answer:
798 To determine how many smaller bags Jim found, we need to follow these
799 steps:
800 1. **Calculate the total amount of gold coins Jim finds over 8 hours:**
801 Jim finds 25 gold coins per hour for 8 hours. The total gold coins he
802 finds is: $[25 \times 8 = 200 \text{ \textit{ gold coins}}]$
803 $[25 \times 8 = 200 \text{ \textit{ gold coins}}]$
804 2. **Determine the amount of gold coins in the smaller bags:** The
805 problem states that each smaller bag has half as much gold as a treasure
806 chest. Since a treasure chest contains 100 gold coins, a smaller bag will
807 have: $[\frac{100}{2} = 50 \text{ \textit{ gold coins}}]$
808 $[\frac{100}{2} = 50 \text{ \textit{ gold coins}}]$
809

810 3. ****Set up an equation to find the number of smaller bags (x) **** The
811 total amount of gold coins found in the smaller bags over 8 hours should
812 equal the total gold coins found minus the gold coins in the treasure
813 chest. Let (x) be the number of smaller bags found. The total gold
814 coins found in the smaller bags is: $50x$ This total must equal the
815 remaining gold coins after subtracting the treasure chest: $50x = 200 -$
816 100 $50x = 100$
817 4. ****Solve for (x) **** To find (x) , divide both sides of the
818 equation by 50: $x = \frac{100}{50} = 2$ Thus, the correct answer is:
819 $\boxed{2}$ Final Answer: $\boxed{2}$
820 Final Answer: 2
821 **### student answer:**
822 To find the number of smaller bags Jim found, we first calculate the
823 total amount of gold coins he has.
824 Since he finds 25 gold coins per hour for 8 hours, the total amount of
825 gold coins he finds is $25 \times 8 = 200$ gold coins. Next, we determine
826 the amount of gold coins in each smaller bag. Each smaller bag has half
827 as much gold as a treasure chest, which is $\frac{100}{2} = 50$ gold
828 coins.
829 Now, we divide the total amount of gold coins found by the amount in each
830 smaller bag to find the number of smaller bags: $\frac{200}{50} = 4$.
831 Therefore, Jim found $\boxed{4}$ smaller bags. Final Answer: $\boxed{4}$.
832 Final Answer: 4
833 **### score:**
834 0

835
836
837 -----similar such examples from the validation set-----
838

839 First, carefully analyze the student's answer. Compare it to the
840 teacher's answer and identify any mistakes or areas where the student's
841 reasoning could be improved. Consider the following:
842

- 843 1. Did the student understand the problem correctly?
- 844 2. Did they use the right approach to solve the problem?
- 845 3. Are there any calculation errors?
- 846 4. Is their reasoning clear and logical?
- 847 5. Did they miss any important steps?

848

849 Next, craft a new answer that addresses the student's misunderstandings
850 or reinforces correct thinking. Your new answer should:
851

- 852 1. Use clear, step-by-step reasoning
- 853 2. Explain any concepts the student may have misunderstood
- 854 3. Provide additional context or examples if necessary
- 855 6. Use the same calculation format as the teacher's answer. If the
- 856 teacher answer involves latex and involves terms like $\frac{1}{2}$, π make
- 857 sure to extract the answer with the necessary latex keywords.
- 858 5. All the teacher's final answers end inside " $\boxed{\quad}$ ". Ensure, that
- 859 your answers also follow this format.
- 860 6. Lead to the correct final answer

861
862 Write your new answer using the following format:
863

new_answer

864 [Step-by-step reasoning with calculations in the format shown above]
865 Final Answer: [Correct numerical/latex answer]
866

867
868 Remember, your goal is to help the student learn and improve their
869 problem-solving skills. Focus on explaining the reasoning clearly and
870 addressing any specific issues in the student's original answer.
871

872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917