

First-Step Advantage: Importance of Starting Right in Multi-Step Math Reasoning

Anonymous ACL submission

Abstract

Language models can solve complex reasoning tasks better by learning to generate rationales for their predictions. Often these models know how to solve a task but their auto-regressive decoding nature leads to incorrect results if they start incorrectly. We observe that smaller models in particular, when corrected, can solve a task that they would have otherwise struggled with. We demonstrate this phenomenon by using a larger model to guide smaller models, which leads to significantly improved performance (up to +24 points on the GSM8K dataset by 7B models). To assist smaller models in initiating the starting step, we propose QuestCoT, where a smaller model first *asks itself how to start*, before proceeding with a chain of reasoning. On various multistep mathematical reasoning datasets over multiple smaller models, we show that getting the right start can lead to significant performance gains across all models (gains of up to +6 points on GSM8K, +9 on SVAMP, +5 on ASDiv, and +7 on MultiArith).

1 Introduction

Over the years, large language models (LLMs) have improved their reasoning abilities by explaining their intermediate thoughts (Wei et al., 2022b). This trend has been extended to smaller models¹, either through pre-training (Jiang et al., 2023; Magnusson et al., 2023), fine-tuning (Yu et al., 2023; Shao et al., 2024), or knowledge distillation (Shridhar et al., 2023b; Yuan et al., 2023; Magister et al., 2023; Hsieh et al., 2023). While it is commonly assumed that smaller models acquire new knowledge through fine-tuning or distillation, recent research by Gekhman et al. (2024) suggests that the acquisition of new knowledge is quite slow. Instead, models often improve in the areas they are already familiar with. This suggests that while models

¹we use smaller models in a relative sense and most of our experiments are carried out on models smaller or equal to 7B parameters

may have the knowledge to solve a given task, they struggle to understand how to apply it effectively.

Wang et al. (2023b) demonstrates that model accuracy improves significantly when multiple reasoning chains are generated, indicating that the model understands how to answer the given problem. However, models often struggle to select the correct initial chain, and if they start on an incorrect reasoning path, it becomes difficult to fix it due to the autoregressive nature of decoding. Similarly, in our work, we observed that if a smaller model initiates an incorrect reasoning chain, it will continue down that incorrect path. Conversely, if the initial step is correctly determined, the model can successfully complete tasks that it would otherwise find challenging.

In this work, we first investigate whether providing initial guidance can improve the reasoning capabilities of smaller language models. We then investigate whether the quality of this initial guidance varies depending on the expertise of different large language models (LLMs). In particular, we investigate whether smaller models can use this guidance without fine-tuning or additional training, and whether models of different sizes benefit equally. Finally, we investigate whether the benefits of initial guidance extend beyond simple two-step problems to tasks requiring four to eight steps of reasoning.

Once the critical role of initial step guidance in reasoning is established, we focus on enabling smaller models to learn *how to start correctly*. To this end, we introduce QuestCoT, a self-questioning guidance mechanism designed to teach models *how to start*. With QuestCoT, the model first generates a sub-question that initiates the reasoning chain, and then follows that path. Essentially, it identifies the most effective reasoning chains needed to answer the given question. A comparison of our proposed methodology, QuestCoT and Chain-of-Thought (CoT) is demonstrated in

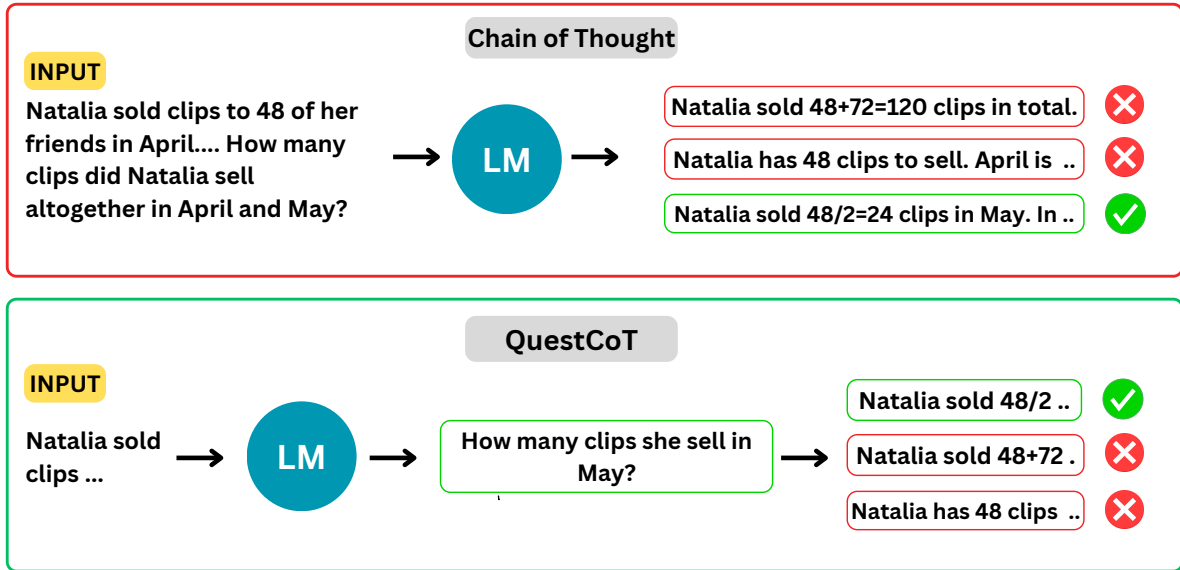


Figure 1: **Comparison between Chain-of-Thought (CoT) approach and QuestCoT.** The CoT approach enables a Language Model (LM) to generate accurate answers through multiple samplings, yet it frequently struggles to confidently select the correct one. Conversely, QuestCoT utilizes self-question-guided generation, which facilitates the model’s ability to choose the appropriate reasoning chain with higher confidence.

Figure 1.

We demonstrate the importance of self-questioning for initializing reasoning chains (QuestCoT) on several mathematical datasets involving multi-step word problems. Consistent performance improvements were observed for all smaller models (all within 7B parameters). Moreover, QuestCoT performs similarly to expert LLM guidance improving the quality of reasoning and outperforms the standard reasoning techniques of chain-of-thought (Wei et al., 2022b, CoT) and sub-question decomposition approaches (Shridhar et al., 2022; Zhou et al., 2023, Subques).

2 Related Work

It is possible to elicit reasoning abilities from LLMs through in-context learning, either by providing the model with intermediate steps (Wei et al., 2022b; Kojima et al., 2023; Yang et al., 2023; Wang et al., 2023b), or by decomposing the problem into smaller sub-problems (Shridhar et al., 2022; Zhou et al., 2023) and solving them to reach the final answer. However, if the problem is misinterpreted, it can lead to a cascade of errors in subsequent steps.

To counter this, several techniques have been proposed to intervene and correct intermediate steps by providing feedback on their own generations, and eventually “self-correcting” their own generations (Welleck et al., 2022; Madaan et al.,

2023; Shridhar et al., 2023a). While the LLM’s ability to revise its own generations may prove helpful in many cases, it sometimes leads to worse results in refinement, requiring a “rollback” to the previous output (Shridhar et al., 2023a). To address this, (Yao et al., 2023) introduces the Tree of Thoughts (ToT), which plans subsequent steps to solve a reasoning task (Huang et al., 2022; Wang et al., 2023a,c). ToT conceptualizes the decision-making process as a series of heuristically based decisions. Through deliberate search, ToT explores different reasoning paths and self-reflects on its decision at each step. We, on the other hand, propose to get the first step right, thus reducing the cost of “finding” and “fixing” errors.

Previous work has also focused on understanding *when* to intervene and correct the errors. Saha et al. (2023) presented an approach based on Theory of Mind (Kosinski, 2023; Kadavath et al., 2022), where a teacher model intervenes in a student model only for harder questions by creating an implicit mental model of the student’s understanding. In contrast, an alternative that avoids the need to backtrack and correct mistakes, thus saving time and effort, is to *start right*.

3 First-Step Advantage

In this section, we address three research questions: 1) the ability of smaller models to solve a reasoning

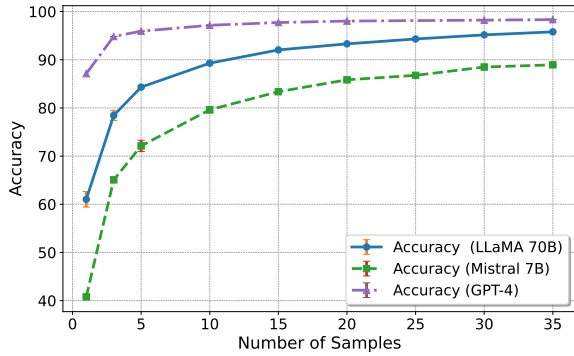


Figure 2: Accuracy (if an answer exists in one of the output chains) comparison on GSM8K data set between different sized models: Mistral 7B, LLaMA-70B, and GPT-4.

task, 2) the importance of taking the correct *first step* in reasoning, and 3) how smaller models can learn to take the correct first step.

3.1 Are smaller models capable of solving a reasoning task?

Hypothesis *Smaller models can solve a given task but are not confident enough to choose the correct reasoning chain.*

For multi-step reasoning tasks, performance generally improves with increasing model size (Wei et al., 2022a). While this trend is generally observed, we hypothesize that smaller models can solve reasoning tasks (beyond what their maj@1 accuracy indicates), but often fail to choose the correct initial chain.

Experimental Design We investigate the ability of smaller models to solve reasoning tasks by sampling their output chains multiple times [1, 3, 5, 10, 15, 20, 25, 30, 35]. A temperature setting of 0.7 is used to generate diverse multiple samples. We compared the performance of the smaller model (Mistral-7B (Jiang et al., 2023)) with the larger ones (LLaMA2-70B (Touvron et al., 2023), and GPT-4 (OpenAI, 2023)) on the GSM8K dataset (Cobbe et al., 2021) for mathematical reasoning. Our analyses were conducted on a test set of 1,319 samples using a 4-shot Chain-of-Thought (CoT) reasoning chain. The prompts used are listed in the Appendix (Figure 10).

Our approach To assess whether smaller models understand how to solve a problem but fail to select the correct reasoning chain on their first attempt, we generate multiple samples from the model and check whether a correct answer appears in any of

them. This method will show that the model understands how to solve the problem, but has difficulty selecting the correct chain at first.

Results Figure 2 illustrates the accuracy of whether an answer exists in one of the sampled outputs for different model sizes (ranging from Mistral 7B to LLaMA 70B to GPT-4, where the true size of GPT-4 is unknown but presumably large) when sampled multiple times. In a single sample, the performance gap between GPT-4 and Mistral 7B is nearly 50 points, indicating GPT-4’s superior ability to select the correct reasoning chain in its initial sample. However, with 35 samples, this gap narrows to less than 10 points, suggesting that smaller models can answer correctly but struggle to consistently select the right chain in their first few attempts. Conversely, GPT-4’s accuracy does not increase significantly with additional samples, suggesting greater consistency in generating and selecting the correct inference chain. This finding is consistent with Lyu et al. (2024), which suggests that scaling improves output consistency.

Key Findings We observed that smaller models can answer a reasoning question when sampled multiple times, but fail to select the correct reasoning chain on the first attempt.

3.2 Importance of starting right

Hypothesis *Smaller models can solve a given task if they get the first step right*

To evaluate the impact of providing a hint in the form of the first step, we generate this first step using a large language model (LLM) and provide it as guidance to the smaller model. This approach allows us to test the importance of getting the *first step* right.

Experimental setup We investigate whether providing *first-step* guidance can help smaller models get better results. We evaluate smaller models in the 2B - 7B range, namely Gemma-2B (Team et al., 2024), Phi3-mini 3.8B (Abdin et al., 2024), LLaMA2-7B (Touvron et al., 2023), O1Mo-7B (Magnusson et al., 2023), Mistral-7B (Jiang et al., 2023), and Gemma-7B (Team et al., 2024). All the models are instruction-tuned versions except LLaMA2 and Mistral. For guidance coming from LLMs, we use LLaMA2-70B (Touvron et al., 2023), GPT-3.5, and GPT-4 (OpenAI, 2023). We test our hypothesis on the test set of two datasets: GSM8K with 1319 samples and SVAMP (Patel

Question The toothpaste in Anne’s family’s bathroom contains 105 grams. Anne’s dad uses 3 grams at each brushing, her mom uses 2 grams, Anne and her brother use 1 gram each. Each member of the family brushes their teeth three times a day. How many days will it take for the toothpaste to run out?

LLM-based first step guidance Anne and her brother use 1 gram + 1 gram = «1+1=2»2 grams . Anne’s dad uses 3 grams and Anne’s mom uses 2 grams so together they use 2 grams + 3 grams + 2 grams = «3+2+2=7»7 grams/wash. Each day, the family uses 7 grams/wash * 3 washes/day = «7*3=21»21 grams. Using that amount daily, the toothpaste will last 105 grams / 21 grams/day = «105/21=5»5 days. The answer is 5.

Figure 3: LLM-based first step guidance is highlighted in yellow followed by model generation.

Model	GSM8K				SVAMP			
	CoT	LLM Guidance			CoT	LLM Guidance		
	No guidance	LLaMA2-70B	GPT-3.5	GPT-4	No guidance	LLaMA2-70B	GPT-3.5	GPT-4
Gemma-2B	7.50	12.81	16.23	17.84	34.60	36.30	46.70	49.20
Phi3-Mini-3.8B	76.95	<u>75.10</u>	77.39	80.27	86.30	<u>84.20</u>	86.10	87.80
LLaMA2-7B	10.53	19.48	21.00	23.27	38.00	40.10	41.40	48.20
OIMo-7B	13.64	28.20	36.54	37.90	18.60	40.90	46.50	49.90
Mistral-7B	40.25	46.17	48.82	49.50	62.00	65.60	66.80	73.40
Gemma-7B	46.55	52.23	59.43	63.45	70.30	72.10	74.10	78.30

Table 1: Accuracy comparison when the first step is provided by a larger LLM versus the baseline (no first step provided) for a smaller model. The best results are shown in **bold**. Note that when a weaker model provides guidance (LLaMA2-70B performance is worse than Phi3-mini), it hurts the performance (underlined).

et al., 2021) with 1000 samples. Greedy sampling (temperature=0) was used for sampling and acc@1 accuracy is reported.

Our approach We use large language models (LLMs) to generate the first step of the solution to a given problem by providing specific instructions (details in the Appendix Figure 9). Although the first-step guidance varies by task, for mathematical reasoning tasks, we provide the first step until we encounter a mathematical equation. We perform sanity checks to ensure that no answer is revealed in this step (detailed analysis in section 4) and limit the solution to a maximum of one equation. Since the problem requires at least two to eight equations to solve, the first-step guidance does not lead directly to the answer but provides a solid starting point for the model. Smaller models then decode the answer by following this first-step guidance, and their final accuracy is compared to the baseline without first-step guidance. Figure 3 shows an example of LLM-based first-step guidance (highlighted in yellow).

Results Table 1 demonstrates the usefulness of the first-step guidance provided by LLMs. The performance of the pre-trained models increases by more than 2-3X when a larger model such as GPT-4 is used for first-step guidance. For example, the performance of Gemma-2B (Team et al., 2024) and LLaMA2-7B model (Touvron et al., 2023) goes

from 7.5 → 17.8 and 10.5 → 23.2, respectively, while for OIMo-7B it goes from 13.6 → 37.9 (an almost 3X jump). Performance increases monotonically with larger and more expert models providing first-step guidance (for Gemma-2B, performance increases from 7.5 → 12.8 with LLaMA-70B first-step guidance and to 16.2 with GPT-3.5). For the more expert models on the GSM8K task, Mistral-7B (Jiang et al., 2023) gains almost +10 points (40.25 → 49.50), Gemma-7B gains +17 points (46.5 → 63.4), and Phi3-Mini (Abdin et al., 2024) gains almost +4 points (76.9 → 80.2). The trend is similar for the SVAMP dataset, where accuracy goes from 34.6 → 49.2 for Gemma and 38.0 → 48.2 for LLaMA2-7B with GPT-4 first-step guidance, and from 18.6 → almost 50 for OIMo-7B. This shows the importance of the first-step guidance in arriving at the correct final answer.

Key Findings We observed that when the first step of the reasoning chain is provided to the smaller models as a starting point, the performance of the smaller models improves significantly.

3.3 QuestCoT: Teaching smaller models to get the first step right

Hypothesis Can smaller models learn to get the first step right?

Given that smaller models can get better results if they learn to start right, can we teach smaller models to learn the first step on their own?

Question Gene is sewing a quilt out of old souvenir t-shirts. He has one shirt from each vacation he has been on. Every shirt is its own quilt block. Each row is made of blocks from a different year of vacations. He goes on four vacations a year and has been vacationing since he was 23 years old. He is now 34. How many quilt blocks does he have in total?

CoT He has been on 4 vacations a year for 34 years so he has been on $4*34 = \ll 4*34=136 \gg 136$ vacations. Each vacation is a quilt block so he has 136 quilt blocks. Each quilt block is one shirt so he has 136 shirts. The answer is 136. ✗

QuestCoT How many years Gene has been on vacation? Gene has been on $34 - 23 = \ll 34-23=11 \gg 11$ years of vacations. Each row is made of 4 blocks, and she has 11 rows of blocks. So he has $11*4 = \ll 11*4=44 \gg 44$ blocks in total. The answer is 44. ✓

Figure 4: Example of a comparison between CoT reasoning and QuestCoT. QuestCoT first asks a question that helps to decide the first step and is highlighted in pink.

Model	Dataset							
	GSM8K		SVAMP		ASDiv		MultiArith	
	CoT	QuestCoT	CoT	QuestCoT	CoT	QuestCoT	CoT	QuestCoT
Gemma-2B	7.50	8.76 (↑+1.1)	34.60	35.00 (↑+0.4)	42.34	42.95 (↑+0.6)	17.77	18.88 (↑+1.1)
Phi3-Mini-3.8B	76.95	78.92 (↑+2.0)	86.30	88.40 (↑+2.1)	80.82	82.34 (↑+1.5)	98.83	99.44 (↑+0.6)
LLaMA2-7B	10.53	15.10 (↑+4.5)	38.00	41.10 (↑+3.1)	41.43	40.90 (↓-0.5)	25.55	28.88 (↑+3.3)
OIMo-7B	13.64	19.40 (↑+5.8)	18.60	27.20 (↑+8.6)	39.37	44.40 (↑+5.0)	20.00	27.22 (↑+7.2)
Mistral-7B	40.25	45.47 (↑+5.2)	62.01	65.15 (↑+3.1)	54.18	57.26 (↑+3.0)	61.66	65.55 (↑+3.9)
Gemma-7B	46.55	48.21 (↑+1.6)	70.30	71.40 (↑+1.1)	68.59	69.84 (↑+1.2)	79.44	78.22 (↓-1.2)
LLaMA3-8B	78.86	79.80 (↑+1.0)	83.70	84.89 (↑+1.2)	73.88	74.27 (↑+0.4)	97.77	98.33 (↑+0.5)

Table 2: Accuracy comparison between the chain of thought (CoT) and QuestCoT. QuestCoT achieves the best results across all model sizes for various multi-step mathematical reasoning datasets.

Experimental setup We explore the effect of *starting right* on four multi-step mathematical datasets: GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), ASDiv (Miao et al., 2020), and MultiArith (Roy and Roth, 2015). GSM8K consists of grade-school math word problems with a test set of 1319 samples, requiring between two and eight steps to solve. SVAMP consists of 1000 samples of math word problems designed to challenge systems that require reasoning beyond shallow approaches. ASDiv consists of 2,305 test samples of word problems that were constructed to have more lexical diversity than other datasets at the time. MultiArith is a dataset of 180 test samples published with the algorithmic solver for mathematical word problems.

We tested smaller models ranging from 2B to 8B parameters, starting with Gemma-2B, followed by Phi3-mini with 3.8B parameters, followed by Mistral-7B, LLaMA2-7B, OIMo-7B, and Gemma-7B with 7B parameters, and finally LLaMA3-8B with 8B parameters. We report the top-1 accuracy (maj@1) on the test sentences of both datasets. To compare CoT and QuestCoT, we used 4-shot prompting with prompts randomly selected from the test set. All models were evaluated using a greedy approach (temperature=0, top p=1). A comparison of prompts between CoT and QuestCoT

can be found in the Appendix (Figure 11).

Our approach To help the smaller models learn how to start with a correct first step, we propose an initial question-guided strategy called QuestCoT. With QuestCoT, a model first asks the most important question that will help it start the reasoning chain and then continues that chain. The initial question it asks can also be thought of as a search strategy that looks for the right starting chain and, once selected, continues along that path. A comparison with CoT and QuestCoT is presented in Figure 4. Note that the model learns this questioning itself, and the only change from CoT is to add an extra question in the prompt as a demonstration.

Results We test the effectiveness of QuestCoT against one of the most popular reasoning strategies: CoT. QuestCoT outperforms CoT on all four datasets for all models except LLaMA2-7B on ASDiv and Gemma-7B on MultiArith. Smaller models such as Gemma-2B and Phi-mini-3.8B gain between +0.5 and +2 points on all four datasets. We hypothesize that Gemma-2B’s limited gains are due to its initial weak performance and under-training, while Phi3-mini is already a very strong model with performance in the 80s and 90s, making further improvement difficult. Nevertheless, improvements are observed in both cases.

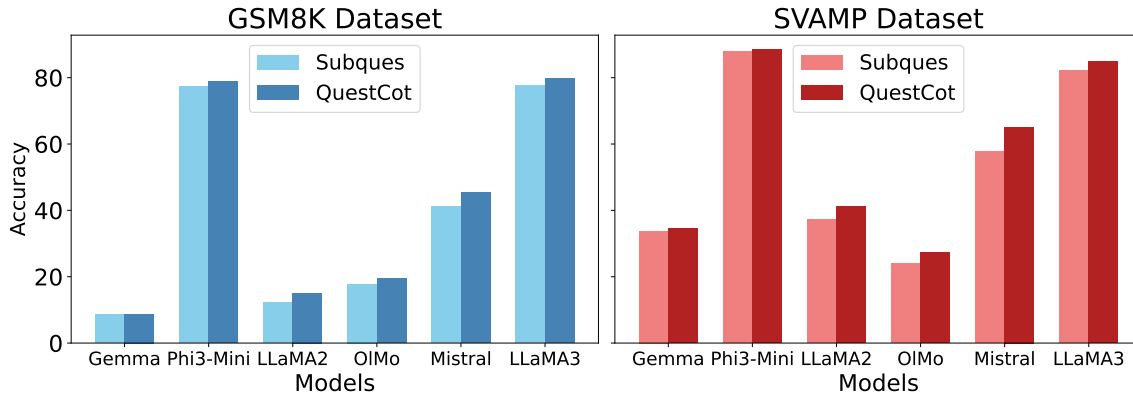


Figure 5: Accuracy comparison between Subques and QuestCot on the GSM8K and SVAMP datasets. Gemma refers to Gemma-2B, Phi3-Mini is Phi3-mini-3.8B, and LLaMA2, OIMo, and Mistral are all 7B variants, while LLaMA3 is LLaMA3-8B.

332 Performance improves significantly with the 7B
 333 models, with OIMo-7B showing the most gains
 334 (+6 on GSM8K, +9 on SVAMP, +5 on ASDiv,
 335 and +7 on MultiArith). This is followed by
 336 LLaMA2-7B and Mistral-7B, which show gains of
 337 +3 – 5 points, and Gemma-7B, which shows gains
 338 of +1 – 2 points. Similar to Phi3-mini, LLaMA3-
 339 8B’s baseline performance is quite high, showing
 340 gains of +0.5 – 1 points.

341 **Key Takeaways** Smaller models improve their
 342 performance by learning to get the first step right by
 343 asking themselves how to start. This improvement
 344 is achieved with our proposed approach, QuestCoT.

345 4 Analysis

346 **Does the first step leak the final answer?** We
 347 investigate whether the performance gains from
 348 LLM guidance are due to LLMs leaking the answer
 349 to the smaller models. To verify this, we created a
 350 development set of 1000 samples from the GSM8K
 351 training set. By comparing the generated first-step
 352 answers with the final answers in the dataset, we
 353 found that in 999 out of 1000 samples, the answers
 354 did not match. Furthermore, our instructions to
 355 the LLMs specified that they could only generate
 356 the first step, corresponding to the first step in the
 357 inference chain with only the first equation, and
 358 could not reveal the final answer. This strategy was
 359 applied consistently across all data sets. Since each
 360 question required at least 2-8 steps to solve, we are
 361 confident that the final answer was not revealed.
 362 Furthermore, if the approach relied on revealing
 363 the final answer, the QuestCoT approach would not
 364 have been effective in the prompt style at all.

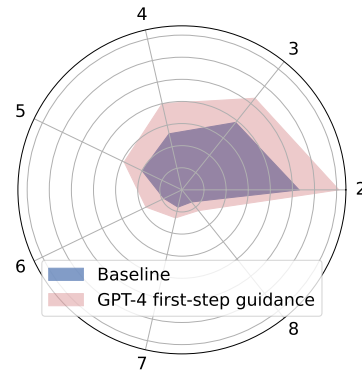


Figure 6: Accuracy comparison between baseline (no guidance) and LLM guidance (GPT-4) for the Mistral-7B model on the GSM8K dataset. 2-8 represents the number of steps required to solve the problem.

365 **Can first-step guidance go beyond two-step**
 366 **problems?** Figure 6 illustrates the performance
 367 of the Mistral-7B model with and without first-step
 368 LLM guidance for different steps in the GSM8K
 369 dataset. For all steps (2 to 8), first-step guidance im-
 370 proves performance, suggesting that starting with a
 371 solid foundation can help reasoning over a longer
 372 context.

373 **What if subquestions are included at each step?**
 374 The subquestion that guides the model on *how to*
 375 *start* can be applied to any reasoning step in the
 376 chain to guide that specific step. This approach
 377 is similar to subquestion decomposition (Shridhar
 378 et al., 2022; Zhou et al., 2023, Subques), where a
 379 complex reasoning problem is first broken down
 380 into simpler problems that are then solved sequen-
 381 tially. Figure 5 shows a comparison between Sub-
 382 ques and QuestCoT over different models on two

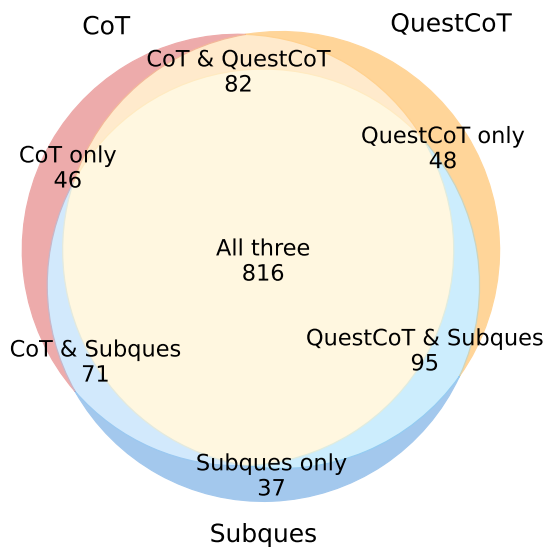


Figure 7: Venn diagram to show when different strategies got the solutions right.

383 datasets: GSM8K and SVAMP. QuestCoT shows
 384 higher accuracy across all models on both datasets
 385 while incurring lower token costs. Through manual
 386 inspection of over 100 samples, we found that
 387 introducing additional questions increases the like-
 388 lihood of errors propagating through the inference
 389 chain. In contrast, QuestCoT guidance avoids this
 390 by focusing solely on *how to start*. These errors
 391 can result from incorrect decomposition, incorrect
 392 reasoning about the decomposed problem, or in-
 393 consistencies throughout the inference chain.

394 **Comparing CoT, Subques and QuestCoT** Fig-
 395 ure 7 presents a Venn diagram illustrating when
 396 different strategies correctly solve problems on
 397 the GSM8K dataset using the Phi-3 mini-model.
 398 QuestCoT has more overlaps with both CoT and
 399 Subques (82 and 95, respectively) than the overlaps
 400 between CoT and Subques (71). This suggests that
 401 while CoT and Subques each have unique cases
 402 where they are successful, QuestCoT captures the
 403 strengths of both strategies, resulting in better over-
 404 all performance.

405 **Can QuestCoT work with even smaller models?**
 406 We tested our approach on the O1Mo-1B model,
 407 which has 1 billion parameters, and found that
 408 it was not well equipped to understand the in-
 409 structions or to generate a reasoning chain start-
 410 ing with an initial question (a necessary condition
 411 for QuestCoT). On the GSM8K dataset, the CoT
 412 performance was 3%, and QuestCoT performed
 413 comparably at 3.5%, with the outputs not look-

414 ing significantly different. As a result, we did not
 415 observe any statistically significant improvements.
 416 We suspect that because the O1Mo-1B model’s CoT
 417 abilities are quite limited in mathematical reason-
 418 ing tasks, it cannot leverage any advantages from
 419 QuestCoT.

420 5 Deeper exploration of why QuestCoT 421 works

422 Understanding why one technique outperforms an-
 423 other can be quite challenging. To address this, we
 424 examined instances where QuestCoT was success-
 425 ful and conducted a detailed analysis. We identified
 426 situations where CoT failed and broadly categor-
 427 ized these errors. Here are some error types where
 428 QuestCoT was beneficial:

429 **Unnecessary calculations** A common mistake
 430 CoT makes is performing unnecessary calculations
 431 on the numbers in the statement. These numbers
 432 may be completely irrelevant to the problem, or
 433 they may need to be used in a different way than
 434 the model uses them. QuestCoT helps to correct
 435 these errors by initiating the reasoning process with
 436 an appropriate question. An example of an unnec-
 437 essary calculation is given in Figure 8 with the
 438 main error shown in red, where there was no need
 439 to calculate the total cost of coffee for the entire
 440 week at the old price.

441 **Real-world knowledge** Often, the first step
 442 in CoT seems somewhat "rushed," focusing on
 443 quickly manipulating numbers without consider-
 444 ing real-world facts or knowledge. In these cases,
 445 the model demonstrates its understanding of these
 446 facts and knowledge in the subsequent steps but
 447 cannot elicit it immediately in the first step. This
 448 suggests that encouraging the model to think more
 449 deliberately in the first step (e.g., by allowing it
 450 to consider what needs to be done before it starts
 451 reasoning) may remedy this problem. These sce-
 452 narios illustrate the effectiveness of QuestCoT. An
 453 example is shown in Figure 8, where the model
 454 fails to convert "half a dozen" to 6, and instead con-
 455 tinues its calculations with 0.5 (as shown in blue).
 456 Although the model demonstrated its understand-
 457 ing of "dozen" later in the problem, since it started
 458 incorrectly, it was unable to correct the chain later.

459 **Context understanding** With CoT, the model
 460 often confuses or misses the context in the problem
 461 statement and makes incorrect initial assumptions
 462 that are difficult to recover from in later steps. For

Unnecessary calculations

Question Roger goes to the store to buy some coffee. The normal brand of coffee he buys cost \$5 per pound. He had to buy a more expensive brand that cost 20% more since his favorite brand was sold out. He decides to buy a week's worth of coffee and he uses 1 pound of coffee per day. He also decided to buy himself a donut for \$2. How much did everything cost?

CoT Roger buys 7 pounds of coffee at \$5 per pound, so the cost of coffee is $7 * \$5 = \35 . The more expensive brand of coffee costs 20% more, so it costs $\$5 * 1.20 = \6 per pound. The cost of the more expensive coffee is $7 * \$6 = \42 . Roger also buys a donut for \$2. The total cost of everything is $\$42 + \$35 + \$2 = \79 . Final Answer: 79 ✗

QuestCoT What is the cost of the more expensive brand of coffee? The more expensive brand costs 20% more than the normal brand, so it costs $\$5 * 20/100 = \1 more per pound. So, the more expensive brand of coffee costs $\$5 + \$1 = \$6$ per pound. Roger buys a week's worth of coffee, which is 7 days, and he uses 1 pound of coffee per day. So, he buys 7 pounds of the more expensive coffee. The total cost of the coffee is $7 \text{ pounds} * \$6/\text{pound} = \42 . He also buys a donut for \$2. So, the total cost of everything is $\$42$ (coffee) + $\$2$ (donut) = $\$44$. Final Answer: 44 ✓

Real world knowledge

Question Twenty dozen cups cost \$1200 less than the total cost of half a dozen plates sold at \$6000 each. Calculate the total cost of buying each cup.

CoT Let's find the total cost of half a dozen plates: $6000 * 0.5 = \$3000$ Now, let's find the total cost of twenty dozen cups: $\$3000 + \$1200 = \$4200$. Since there are 12 cups in a dozen, there are $20 * 12 = 240$ cups in twenty dozen. The cost of each cup: $\$4200 / 240 = \17.50 . Final Answer: 17.50 ✗

QuestCoT How many cups are in 20 dozen? 1 dozen is equal to 12 cups, so 20 dozen is equal to $20 * 12 = 240$ cups. Half a dozen plates is equal to $0.5 * 12 = 6$ plates. The total cost of half a dozen plates is $6 * \$6000 = \36000 . The total cost of buying 20 dozen cups is \$1200 less than the cost of half a dozen plates, so it is $\$36000 - \$1200 = \$34800$. The total cost of buying 20 dozen cups is \$34800, so the cost of each cup is $\$34800 / 240 = \145 . Final Answer: 145 ✓

Context understanding

Question You can buy a movie super ticket for \$20 that includes rights to watch any movie and you get one popcorn with a soda for only \$1 extra. Or you can buy a movie ticket for \$12 and buy one popcorn and a soda at regular price. What's the regular price of popcorn if the soda costs \$3 and you didn't buy the super ticket and ended up saving \$2?

CoT Let's denote the regular price of popcorn as P. If you buy the regular ticket and popcorn and soda, the total cost would be: $\$12$ (movie ticket) + P (popcorn) + $\$3$ (soda) = $\$12 + P + \3 . If you buy the super ticket and popcorn with soda, the total cost would be: $\$20$ (super ticket) + $\$1$ (popcorn) + $\$3$ (soda) = $\$20 + \$1 + \$3$. You saved \$2 by not buying the super ticket, so the cost of the regular ticket and popcorn and soda is \$2 more than the cost of the super ticket and popcorn and soda: $\$12 + P + \$3 = \$20 + \$1 + \$3 + \2 . Now, let's solve for P: $\$12 + P + \$3 = \$20 + \$1 + \$3 + \2 . $\$12 + P + \$3 = \$26$. $P + \$15 = \26 . $P = \$26 - \$15 = \$11$. Final Answer: 11 ✗

QuestCoT What is the total cost of buying a ticket, popcorn and soda? Let's denote the regular price of popcorn as P. The total cost of buying a regular ticket, popcorn, and soda is $\$12 + P + \$3 = \$15 + P$. The total cost of buying a super ticket, popcorn, and soda is $\$20 + \$1 = \$21$. Since you saved \$2 by not buying the super ticket, the cost of the regular ticket, popcorn, and soda is $\$21 - \$2 = \$19$. Therefore, $\$15 + P = \19 . Solving for P, we get $P = \$19 - \$15 = \$4$. Final Answer: 4 ✓

Figure 8: Example cases associated with different error types and how QuestCoT assists in getting the correct reasoning chain.

463 example, in Figure 8 we can see that despite follow-
464 ing a fairly elaborate reasoning template of variable
465 assumptions, the CoT reasoning misses the fact that
466 the price of the Super Ticket already includes the
467 price of the popcorn. The incorrect assumption is
468 highlighted in the response.

469 **Other errors** Other errors we have observed in-
470 clude that QuestCoT may be better at handling di-
471 rect numeric computations and understanding the
472 simple arithmetic required by the problems. In
473 contrast, CoT may deviate or fail to capture the
474 essential computational aspects of the query. In
475 addition, CoT sometimes takes more steps than
476 necessary, resulting in an incorrect final solution.

6 Conclusion

477
478 We find that smaller models sometimes struggle
479 with taking the correct first step, but their perfor-
480 mance increases significantly once this step is cor-
481 rected. We demonstrated this by using LLMs to
482 guide smaller models to take the correct first step,
483 helping them to establish the correct reasoning
484 chain. To facilitate this for smaller models, we pro-
485 pose QuestCoT, which uses initial question-based
486 guidance to improve their reasoning themselves
487 without any guidance. We demonstrate the effec-
488 tiveness of our approach on four multi-step math-
489 ematical reasoning datasets using different open-
490 source small models.

7 Limitations

Our experiments focus only on English datasets, and we have not tested the performance of our methods in other languages. We acknowledge that including a sub-question to initiate the chain of reasoning may incur some additional cost compared to the chain-of-thought approach. However, it is significantly less costly than the sub-question decomposition approach and yields superior performance compared to both methods.

8 Ethical Considerations

The initial guidance provided by expert LLMs or the self-questioning mechanism could introduce or perpetuate bias due to the unknown training process of the large LLMs (especially the closed-source LLMs such as GPT-4). It’s crucial to evaluate and mitigate any biases in the generated output of the LLMs.

References

Marah Abidin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

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](#).

Zorik Gekhman, Gal Yona, Roei Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander J. Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. [Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes](#). *ArXiv*, abs/2305.02301.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. [Inner monologue: Embodied reasoning through planning with language models](#).

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,

L’elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *ArXiv*, abs/2310.06825.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. [Language models \(mostly\) know what they know](#).

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. [Large language models are zero-shot reasoners](#).

Michal Kosinski. 2023. [Theory of mind might have spontaneously emerged in large language models](#).

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2024. [Calibrating large language models with sample consistency](#). *arXiv preprint arXiv:2402.13904*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#).

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. [Teaching small language models to reason](#).

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, et al. 2023. [Paloma: A benchmark for evaluating language model fit](#). *arXiv preprint arXiv:2312.10523*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. [A diverse corpus for evaluating and developing English math word problem solvers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

OpenAI. 2023. [Gpt-4 technical report](#).

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. [Are NLP models really able to solve simple math word problems?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

599	<i>Language Technologies</i> , pages 2080–2094, Online.	Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models . <i>ArXiv</i> , abs/2307.09288.	657
600	Association for Computational Linguistics.		658
601	Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems . In <i>Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing</i> , pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.	Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models .	659 660 661 662 663
606	Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach? teacher explanations improve student performance via personalization . In <i>Thirty-seventh Conference on Neural Information Processing Systems</i> .	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models .	664 665 666 667
611	Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models . <i>arXiv preprint arXiv:2402.03300</i> .	Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023c. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents .	668 669 670 671 672
616	Kumar Shridhar, Harsh Jhamtani, Hao Fang, Benjamin Van Durme, Jason Eisner, and Patrick Xia. 2023a. Screws: A modular framework for reasoning with revisions .	Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models . <i>arXiv preprint arXiv:2206.07682</i> .	673 674 675 676 677
620	Kumar Shridhar, Jakob Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. Automatic generation of socratic subquestions for teaching math word problems .	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models . In <i>Advances in Neural Information Processing Systems</i> .	678 679 680 681 682
624	Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023b. Distilling reasoning capabilities into smaller language models . In <i>Findings of the Association for Computational Linguistics: ACL 2023</i> , pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.	Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khoshabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct .	683 684 685 686
630	Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology . <i>arXiv preprint arXiv:2403.08295</i> .	Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers .	687 688 689
636	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models .	690 691 692 693
637		Long Long Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zheng Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models . <i>ArXiv</i> , abs/2309.12284.	694 695 696 697 698
643		Zheng Yuan, Hongyi Yuan, Cheng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models . <i>ArXiv</i> , abs/2308.01825.	699 700 701 702
644		Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models .	703 704 705 706 707
645			
646			
647			
648			
649			
650			
651			
652			
653			
654			
655			
656			

Below is a math word problem that requires multiple steps to solve it. Your job is to only provide the first step of the solution and not to reveal the final answer. The first step consists of only one equation in it.

Input: Thomas is training at the gym to prepare for a competition. He trained for 5 hours every day for a month (30 days). If he continues to train for the next 12 days, how many hours will he spend on training in total?

Response: **Total hours for first month=5hours/day×30days**

Figure 9: Instructions to generate first step by LLM. The model-generated output is presented in **green**.

Below is an instruction that describes a task, paired with an ### Input that provides further context. Write a ### Response that appropriately completes the request.

Instruction: Solve the given math problem step by step, and put your final answer after 'Final answer:'.

Input: Thomas is training at the gym to prepare for a competition. He trained for 5 hours every day for a month (30 days). If he continues to train for the next 12 days, how many hours will he spend on training in total?

Response: In total Thomas would train on $30 + 12 = \langle 30+12=42 \rangle 42$ days. Thomas trained 5 hours every day, which would bring us to $42 * 5 = \langle 42*5=210 \rangle 210$ hours of training in total. Final Answer: 210 <eot_id>

[Similar 3 more examples randomly sampled from the training set]

Input: Nina made a two-layer cake and a dozen cupcakes for her friend's birthday party. Each layer of cake takes the same amount of sugar to make as a dozen cupcakes. Nina used 720 grams of sugar to bake everything. How many grams of sugar are in one cupcake?

Response: **Model generated response ..**

Figure 10: Four-shot CoT demonstration.

Below is an instruction that describes a task, paired with an ### Input that provides further context. Write a ### Response that appropriately completes the request.

Instruction: Solve the given math problem step by step, and put your final answer after 'Final answer:'.

Input: Thomas is training at the gym to prepare for a competition. He trained for 5 hours every day for a month (30 days). If he continues to train for the next 12 days, how many hours will he spend on training in total?

Response: **How many days will Thomas train in total?** In total Thomas would train on $30 + 12 = \langle 30+12=42 \rangle 42$ days. Thomas trained 5 hours every day, which would bring us to $42 * 5 = \langle 42*5=210 \rangle 210$ hours of training in total. Final Answer: 210 <eot_id>

[Similar 3 more examples randomly sampled from the training set]

Input: TNina made a two-layer cake and a dozen cupcakes for her friend's birthday party. Each layer of cake takes the same amount of sugar to make as a dozen cupcakes. Nina used 720 grams of sugar to bake everything. How many grams of sugar are in one cupcake?

Response: **Model generated response ..**

Figure 11: Four-shot QuestCoT demonstration. The only difference from CoT is **underlined**.