

# SCREWS 🧠 : A Modular Framework for Reasoning with Revisions

Anonymous ACL submission

## Abstract

Large language models (LLMs) can improve their accuracy on various tasks through iteratively refining and revising their output based on feedback. Sometimes these *revisions* can introduce errors, in which case it is better to roll back to a previous result. Further, revisions are typically homogeneous where the same reasoning method that produced the initial answer is used for revisions, which may not correct errors. We present SCREWS, a modular framework for reasoning with revisions, which is comprised of three main modules: *Sampling*, *Conditional Resampling*, and *Selection*, each consisting of sub-modules that can be hand-selected per task. We apply SCREWS for arithmetic word problems and multi-hop question answering tasks with multiple state-of-the-art LLMs, and find that: pursuing a **heterogeneous** mixture of reasoning strategies proves beneficial when revising, and **selection** between the original and revised responses is needed to fix any errors introduced by revision.

## 1 Introduction

Large Language Models (LLMs) have proven effective on a variety of reasoning tasks (OpenAI, 2023). However, the LLM output is not always correct on its first attempt, and it is often necessary to iteratively refine the outputs to ensure that the desired goal is achieved (Madaan et al., 2023; Welleck et al., 2022; Zheng et al., 2023; Paul et al., 2023). These refinement methods assume that subsequent outputs (either by the same model, or by an external model or some tool) lead to better performance. However, there is no guarantee that subsequent versions must be better; as Figure 1 illustrates, refinement can lead to a incorrect solution. This motivates a *Selection* strategy whereby the model can roll back to an earlier output.

In addition, past work on iterative refinement typically assumes a single, fixed reasoning strategy (Welleck et al., 2022; Huang et al., 2022; Madaan

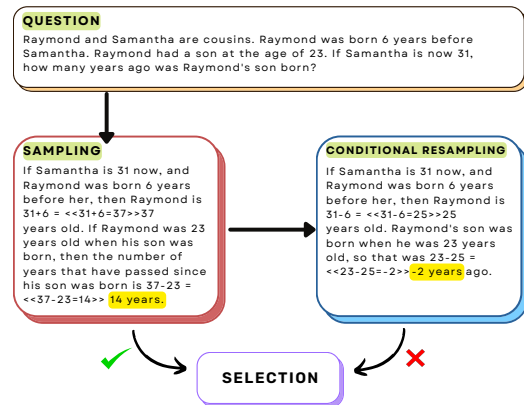


Figure 1: An example demonstrating that *Conditional Resampling* (also known as “refinement”) can lead to incorrect modification of the original answer. The *Selection* module can retract it, if needed.

et al., 2023; Zheng et al., 2023). Humans, however, are more flexible. For example, an inventor may use several strategies to create a new product idea, like brainstorming, reviewing user feedback, and market research, before aggregating them together. Likewise, we take a *modular* approach to answer revisions, allowing us to test different strategies. Different reasoning strategies often have unique strengths and weaknesses. By combining them, one can leverage the advantages of each strategy while compensating for their individual limitations. Heterogenous mixture of strategies can lead to a more robust and comprehensive approach.

In this work, we introduce SCREWS, a modular framework for reasoning with revisions.<sup>1</sup> Figure 2 introduces the three main modules of the framework in detail, namely *Sampling*, *Conditional Resampling*, and *Selection*. For a given task and input sequence, we instantiate SCREWS by fixing the sub-modules for each module (for example, we might select “Chain of Thought” for *Sampling*). The ini-

<sup>1</sup>SCREWS 🧠 stands for “Sampling, Conditional REsampling With Selection.” We will release the code.

063 tial outputs generated by *Sampling* are passed to  
064 *Conditional Resampling*, which decides whether to  
065 generate a revision *conditioned* on the initial sam-  
066 ple, and does so if needed. Finally, all samples and  
067 revisions are given to the *Selection* module, which  
068 selects the best one.

069 We use SCREWS to find the best strategy on two  
070 reasoning tasks: arithmetic reasoning (Cobbe et al.,  
071 2021) and multi-hop question answering (Strate-  
072 gyQA) (Geva et al., 2021). After finding the best  
073 strategy on a held-out set using ChatGPT (based on  
074 Brown et al. (2020)), we verify its effectiveness on  
075 the test set across three LLMs: Llama 2 (Touvron  
076 et al., 2023), ChatGPT, and GPT-4 (OpenAI, 2023).  
077 We find that:

078 **1. Heterogenous sampling** leads to improved  
079 accuracy. While past work typically views self-  
080 refinement as resampling with the same reasoning  
081 strategy, we find that mixed strategies consistently  
082 lead to benefits by complementing each other.

083 **2. Model-based selection** across the samples  
084 and revisions is necessary for good performance.  
085 Otherwise, more of the “refined” revisions are in-  
086 correct than the original predictions, one of the  
087 weaknesses of refinement-style approaches for rea-  
088 soning tasks (Huang et al., 2023).

## 089 2 SCREWS: Methodology

090 In this section, we formally describe SCREWS, our  
091 proposed modular framework for reasoning with  
092 revisions to tackle different reasoning tasks. Given  
093 a problem  $x$ , the goal is to generate an *answer*  
094  $a$ , which in our experiments may be a string or a  
095 number. SCREWS consists of three main modules:  
096 *Sampling*, *Conditional Resampling*, and *Selection*.  
097 Different variants of SCREWS are obtained by in-  
098 stantiating these modules in different ways. The  
099 options for each module are described below and  
100 illustrated schematically in Figure 2. Note that  
101 there are other possible ways to instantiate each  
102 module. However in this work, we study only the  
103 instantiations described below.

104 All of our methods will invoke one or more  
105 stochastic functions, where each function  $\psi$  maps  
106 a tuple of input strings to a *result* string  $y$  that  
107 contains useful information. In practice,  $\psi$  deter-  
108 ministically constructs a prompt from the input  
109 strings and then samples  $y$  from a large pretrained  
110 language model as a stochastic continuation of this  
111 prompt. For a given tuple of input strings, the  
112 prompt constructed for  $\psi$  will typically be a format-

113 ted encoding of this tuple, preceded by a task spe-  
114 cific instruction and several demonstrations (few-  
115 shot examples) that illustrate how  $\psi$  should map  
116 other encoded input tuples to their corresponding  
117 continuations (Brown et al., 2020).

### 118 2.1 Sampling

119 As an example of the simplest and most naive in-  
120 stantiation of the *Sampling* module, given a prob-  
121 lem  $x$ , the model  $\psi$  could directly generate the an-  
122 swer  $y = \psi(x)$  without any intermediate steps. The  
123 value of  $y$  is returned as the answer  $a$  (if there is no  
124 further revision of  $y$ ). We consider the following  
125 more complex instantiations in this work:

126 **Chain of Thought (CoT).** For many reasoning  
127 tasks today, generating explanations improves the  
128 quality of the final answer (Wei et al., 2022; Ko-  
129 jima et al., 2022). Chain of Thought sampling  
130 encourages the model to explain the intermediate  
131 step-by-step reasoning en route to a decision. This  
132 approach is now commonly used in several rea-  
133 soning tasks. We can still define  $y = \psi(x)$ , but  
134 now we expect the prompt continuation to consist  
135 of step-by-step reasoning culminating in the step  
136 by step answer  $y$ , as demonstrated by the few-shot  
137 examples included in the prompt. The answer  $a$   
138 is extracted from  $y$  using a simple deterministic  
139 pattern-matching heuristic.

140 **Sub-question decomposition (Subques).** This  
141 method decomposes the problem  $x$  into simpler  
142 sub-questions  $[x_1, x_2, \dots, x_n]$ . For each sub-  
143 question  $x_i$  in turn ( $i = 1, 2, \dots, n$ ), the model  
144 is called to generate the corresponding sub-answer  
145  $y_i = \psi(x, x_1, y_1, \dots, x_{i-1}, y_{i-1}, x_i)$ . Note that we  
146 generate all questions before seeing any answers;  
147 that choice follows Shridhar et al. (2023), who  
148 found this approach to work better than interleaved  
149 generation of questions and answers. The sequence  
150 of questions may be generated in a single step, ei-  
151 ther by a call to a stochastic function  $\psi_{\text{question}}$ , as in  
152 this work, or by a custom generation module that  
153 has been fine-tuned on human-written questions as  
154 in Cobbe et al. (2021). The answer  $a$  is extracted  
155 from  $y_n$  with a simple heuristic as in CoT.

### 156 2.2 Conditional Resampling

157 The result  $y$  from the *Sampling* module can be  
158 viewed as a *provisional result*,  $y_{\text{curr}}$ . This is passed  
159 to the *Conditional Resampling* module where a  
160 decision is made whether or not to revise it. This is  
161 done in two steps: first deciding whether or not to  
162 revise, and then if so, resampling a new result  $y_{\text{next}}$

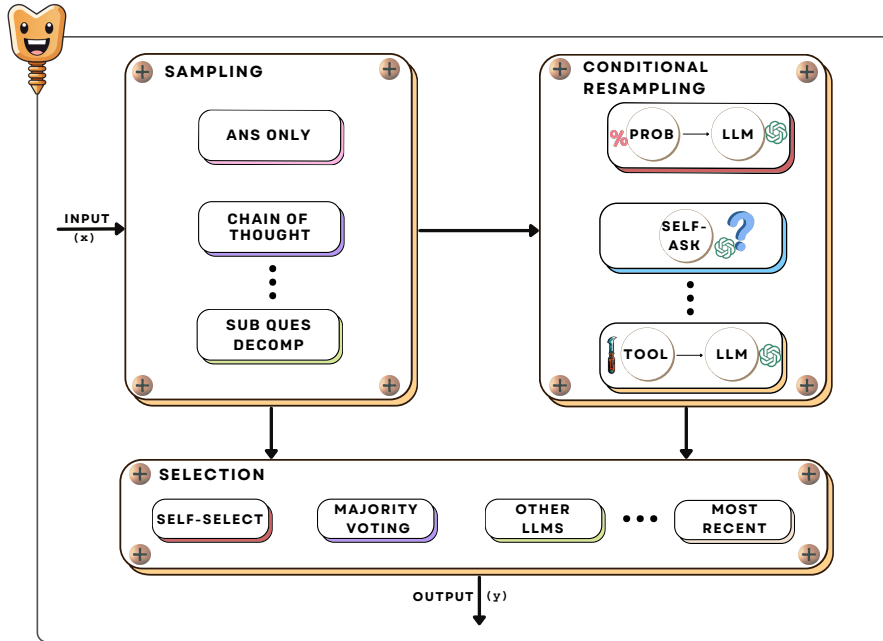


Figure 2: Overview of our modular framework for reasoning with revisions, SCREWS. Each of the three large boxes (“modules”) contains several alternatives (“submodules”). (...) represents other sub-components that can be added to each module, like cached memory or web search for *Sampling*, among others.

using one of the sampling methods mentioned above. The resampling is conditional because  $y_{\text{next}}$  may depend on  $y_{\text{curr}}$ . Our work focuses on the following instantiations for *Conditional Resampling*:

**Self-Ask.** Kadavath et al. (2022) use a function  $\psi_{\text{ask}}(x, y_{\text{curr}})$ . The first token of the result indicates whether  $y_{\text{curr}}$  is correct, for example by starting with “Yes” or “No”. If “Yes”, we do not resample; if “No”, we must resample a revised answer  $y_{\text{next}}$ . In principle, the revision could be iterated as we discuss in Section 5.3, although Kadavath et al. (2022) did not do this.

In our version of self-ask,  $\psi_{\text{ask}}$  is formulated so that  $y_{\text{next}}$  appears in the result string  $\psi_{\text{ask}}(x, y_{\text{curr}})$  following the token “No”. Thus, both steps are efficiently performed by a single call to  $\psi_{\text{ask}}(x, y_{\text{curr}})$ . For this method, we always use greedy decoding (temperature 0) to deterministically select whichever of “Yes” or “No” is more probable.<sup>2</sup>

When the sampling module (Section 2.1) used sub-question decomposition to produce a chain of sub-answers  $y_{\text{curr}} = [y_1, \dots, y_n]$ , rather than

checking and revising only the final result step  $y_n$  by calling  $\psi_{\text{ask}}(x, y_n)$ , we can instead check and revise each step, at the cost of more calls to  $\psi_{\text{ask}}$ . For each provisional sub-answer  $y_i$  in turn (starting with  $i = 1$ ), we predict whether it is correct by calling  $\psi_{\text{ask}}(x, x_1, y_1, \dots, x_{i-1}, y_{i-1}, x_i, y_i)$ . The first time the output is “No”, we resample  $y'_i$  through  $y'_n$ , yielding the revised result  $y_{\text{next}} = [y_1, \dots, y_{i-1}, y'_i, \dots, y'_n]$ . In principle, self-ask could then be applied again at later steps  $> i$  of both the original and revised chains; then choosing among the many resulting chains, using the selection procedures of the next section, would resemble branching in a reasoning tree (Yao et al., 2023).

**Tool use.** For some tasks, we construct  $\psi_{\text{ask}}$  so that it is allowed to use tools (Schick et al., 2023). The reason is that in tasks like fact-checking, it is futile to ask the LLM to check  $y_{\text{curr}}$  because it might not have the requisite knowledge for evaluation. The tools can be used to collect additional information to help the model detect and fix problems in its own generated answer. Tools like search engines or fact retrievers can be used to evaluate correctness and generate a new revision. Other tools like code interpreters are not capable of generating text, but can still be used to evaluate correctness.

<sup>2</sup>A threshold other than 50% could also be selected. Alternatively, the correctness probability of  $y_{\text{curr}}$  could be assessed by a dedicated  $\psi_{\text{check}}(x, y_{\text{curr}})$ , but we were unsuccessful with this as  $\psi_{\text{check}}$  was poorly calibrated, mirroring findings on model calibration (Kadavath et al., 2022; Xiong et al., 2023).

## 2.3 Selection

The last module in SCREWS is the *Selection* module. In this step, we use either a model  $\psi_{\text{select}}$  or simple heuristics to select the *final* result  $y$  from which we then extract the *final* answer  $a$ . In effect, this allows us to construct a simple ensemble of multiple systems.

**LLM-Based Selection.** Just as an LLM was used above to evaluate whether  $y_{\text{curr}}$  is good, an LLM can be used to evaluate whether  $y_{\text{next}}$  is better. We call  $\psi_{\text{select}}(x, y_{\text{curr}}, y_{\text{next}})$  to choose between two result strings.<sup>3</sup> Note that it could be naturally extended to choose between more than two answers. When selection and sampling are implemented using the same LLM, we refer to the method as *self-select* (e.g., in Figure 2).

**Rule-Based Selection.** Many other selection methods can be considered rule-based. Past work on iterative refinement (Madaan et al., 2023; Huang et al., 2022; Zheng et al., 2023) always selects the most recent revision. Majority voting is a simple traditional ensembling method that has been used for selection (Wang et al., 2022; Lewkowycz et al., 2022), but it is costly since it requires several samples.

## 2.4 Related Work

**Sampling.** Prompting LLMs to generate a series of intermediate steps has proven to be effective for improving their reasoning capabilities (Wei et al., 2022; Lewkowycz et al., 2022; Kojima et al., 2022; Wang et al., 2022). Some approaches in this direction include chain of thought (Wei et al., 2022; Zhang et al., 2022; Wang et al., 2022; Lyu et al., 2023) and adding “Let’s think step by step” to the prompt (Kojima et al., 2022). Another approach is “question decomposition”, which decomposes the main problem into simpler problems and solves them iteratively (Min et al., 2019; Shridhar et al., 2022; Zhou et al., 2022; Jhamtani et al., 2023; Radhakrishnan et al., 2023). Each of these approaches has its own advantages depending on the underlying task (Shridhar et al., 2023). However, we are not aware of work combining these methods.

**Conditional Resampling.** The use of feedback to improve generated samples has been well studied, where the feedback can come either from humans (Tandon et al., 2021; Bai et al., 2022; Elgohary et al., 2021), from reward models (Ziegler

et al., 2019; Lu et al., 2022; Shridhar et al., 2022; Christiano et al., 2017; Lightman et al., 2023), from external tools such as code interpreters (Schick et al., 2023; Chen et al., 2022), or from other LLMs (Madaan et al., 2023; Welleck et al., 2022; Fu et al., 2023; Peng et al., 2023; Yang et al., 2022; Zheng et al., 2023; Cohen et al., 2023; Ling et al., 2023; Khalifa et al., 2023). However, even if these feedback mechanisms are infallible, the resulting revisions may introduce new errors. While prior work uses the term “refinement,” we avoid it because refinement implies finer (improved) responses, which is not always the case.

**Selection.** In LLM-based revision, a common selection technique is to select the final result (Madaan et al., 2023; Shinn et al., 2023; Zheng et al., 2023; Yao et al., 2022; Chen et al., 2023; Weng et al., 2022). However, this can lead to accepting incorrect changes made to previously correct results. Other selection methods involve ranking multiple sampled outputs (Burgess et al., 2005; Cobbe et al., 2021) or majority voting (Wang et al., 2022; Lewkowycz et al., 2022; Zheng et al., 2023). These methods often use a homogeneous sampling strategy with changes in hyper-parameters. Our work extends the strategy to heterogeneous sampling and selection.

In Appendix A, we annotate Figure 2 with how several of these prior works can be instantiated within our framework and can be further combined with our proposed strategies.

## 3 Experiments

We use the SCREWS framework to investigate the research questions: 1) what is the best way to conditionally resample; i.e. should we use a mixture of reasoning strategies?; and 2) what is the impact and importance of selection across revisions?

### 3.1 Tasks

We use the framework described by SCREWS on two reasoning datasets: GSM8K (Cobbe et al., 2021) for arithmetic reasoning and StrategyQA (Geva et al., 2021) for multi-hop question answering. GSM8K is a dataset of grade-school-level math word problems with a test set of 1319 samples, each requiring two to eight steps to solve. StrategyQA is a dataset for question-answering where answering each questions typically requires answers to an implicit set of several sub-questions.

Following Magister et al. (2023) and Shridhar

<sup>3</sup>We found the order of  $y_{\text{curr}}$  and  $y_{\text{next}}$  in the prompt was unimportant, so we randomized it in our study.

Sampling	Acc.	Resample %	Conditional Resampling	Acc.	Selection Acc.
<b>GSM8K</b>					
CoT: Chain of thought	76.40	18	CoT	75.70	76.60
			Subques	<u>76.80</u>	<u>77.20</u>
Subques: Subquestion decomposition	75.60	22	CoT	<u>76.30</u>	<u>77.40</u>
			Subques	73.50	<u>76.20</u>
<b>StrategyQA</b>					
CoT: Chain of thought	76.75	34	CoT	74.85	76.25
			+Facts	<u>78.75</u>	<u>79.05</u>
			Subques	76.85	77.25
			+Facts	<b>80.25</b>	<b>80.45</b>

Table 1: The improvements achieved on the dev set by using *Conditional Resampling* and *Selection* for the GSM8K and StrategyQA dataset using ChatGPT model. The fraction of examples resampled is also reported. Underline indicates statistical significance with  $p < 0.05$  compared to the baseline while **Bold** represents the best overall results. +Facts represents using facts alongside the resampling strategy (tool usage).

et al. (2023), we test on the first 490 samples from the training set of StrategyQA (since their test set is unlabeled and we followed the 80-20 train-test split). We also randomly draw 1,000 samples from train set for GSM8K and 400 samples for StrategyQA for the dev set. The demonstration examples for our various stochastic functions  $\psi$  were drawn randomly from the rest of the training set.

Both dataset releases already include subquestions. StrategyQA provides human-annotated oracle subquestions and related facts that can assist in answering the main question (which we use for tool-based conditional resampling described in Section 2.2). In the GSM8K dataset, subquestions were generated by a fine-tuned GPT-3 model and correspond to the steps in a particular correct CoT solution: we will use these for ablation studies (Section 5.2).

### 3.2 Experimental Setup

We always report exact-match accuracy: the percentage of examples on which our final answer  $a$  matches the gold answer. We first use ChatGPT (gpt-3.5-turbo-0613) to determine the best combination of methods on the dev set for each of the tasks. We found the heterogeneous resampling and selection proved to be the best combination and we used it for the test set experiments using three LLMs: ChatGPT again (gpt-3.5-turbo-0613), and GPT-4 (gpt-4-0613), and Llama 2 70B (Touvron et al., 2023). Both ChatGPT and GPT-4 were based on the September 2023 APIs.

**Sampling** We use greedy decoding (temp=0) for all choices of *Sampling* module with 5 provided demonstrations (prompts in Appendix C.1).

**Conditional Resampling** Greedy decoding is

used to first make a binary resampling decision and then to sample. 4-shot prompts (with two correct and two incorrect samples) are used for the GSM8K and StrategyQA datasets (Appendix C.2). For StrategyQA, we use tool-based resampling by including the provided facts from the dataset into the prompt to simulate a (perfect) fact retrieval tool. Even though the facts are provided as part of the dataset, we do not include facts in the prompt when initially calling to  $\psi_{\text{ask}}$  to decide whether to resample, but only when we actually generate  $y_{\text{next}}$ . We studied this scenario because it is more realistic; in practice, tool invocations that retrieve facts, like performing a web search, are expensive and may only be desirable for the difficult examples.

**Selection** For the *self-select* strategy, the prompts (Appendix C.3) include two examples and selection was produced with greedy decoding.

## 4 Results

Table 1 reports the intermediate and final accuracy score after each module of SCREWS, for each of several combinations of module instantiations on the dev set. The best strategies were selected for the test set. Those test scores are reported in Table 2 alongside a Self-Refine (SR) baseline (Madaan et al., 2023), which is akin to always resampling and always selecting the revision (most recent).

### 4.1 Importance of the Selection module

First, in SCREWS, we observe that *Conditional Resampling* does not invariably improve every output. In fact, we see in Table 1 that for some configurations of SCREWS, it often harms the output quality. Because of these regressions, the *Selection* mod-

Dataset	Method	Chain-of-thought			Subquestion Decomposition		
		Llama2-70B	ChatGPT	GPT-4	Llama2-70B	ChatGPT	GPT-4
GSM8K	SR Base	-	74.8	92.9	-	-	-
	+Refine	-	75.0 (↑ 0.2)	93.1 (↑ 0.2)	-	-	-
	Baseline	59.24	76.20	92.18	55.66	77.10	91.24
	+ het <sub>res</sub>	59.46 (↑ 0.4)	76.80 (↑ 0.6)	92.64 (↑ 0.5)	56.28 (↑ 0.6)	77.78 (↑ 0.7)	92.10 (↑ 0.9)
	+ select	59.88 (↑ 0.6)	<u>77.30</u> (↑ 1.1)	<u>93.88</u> (↑ 1.6)	56.64 (↑ 1.0)	<u>78.30</u> (↑ 1.2)	<u>93.44</u> (↑ 2.2)
StrategyQA	Baseline	74.15	75.70	-	73.25	76.10	-
	+ het <sub>facts</sub>	75.65 (↑ 1.5)	<u>77.75</u> (↑ 2.0)	-	<u>75.50</u> (↑ 2.2)	<u>78.10</u> (↑ 2.0)	-
	+ select	<u>75.85</u> (↑ 1.7)	<u>77.95</u> (↑ 2.2)	-	<u>75.65</u> (↑ 2.4)	<u>78.25</u> (↑ 2.1)	-

Table 2: Accuracy on the test set for GSM8K and StrategyQA with chain of thought (left) or subquestion decomposition (right) across three models: Llama 2 70B, ChatGPT and GPT-4. Baseline is the score after initial sampling, het<sub>res</sub> is heterogeneous resampling by changing reasoning strategy, het<sub>facts</sub> is heterogeneous resampling with facts, and select is after *Selection*. Compared to the baseline, underline indicates statistical significance with  $p < 0.05$  and (↑) reports the improvements. SR refers to Self-Refine method, reported by Madaan et al. (2023).

ule is useful: in **all** cases on both the dev and test set, it leads to an increased score over the naive paradigm of retaining the most recently produced revision, implicitly proposed by past work like the SR method.

The cost of LLM-based selection is minor. Recall that in contrast to SR, SCREWS does not always resample; it resamples on only a fraction of the examples (Table 1). Further, selection is relatively inexpensive as few output tokens need to be produced and leads to better performance than SR. Thus, not only is LLM-based selection better than no selection—combined with *Conditional Resampling*, it is also less expensive. Exact token cost is detailed in Appendix B.

## 4.2 Conditional Resampling Works Better with a Method Change

**Heterogenous sampling is effective.** Like to previous findings (Madaan et al., 2023; Huang et al., 2023), we observe that *sampling* and then *resampling* with the same strategy (either chain of thought or subquestion decomposition) leads to performance degradation (76.40 → 75.70 for CoT and 75.60 → 73.50 for Subques) as shown in Table 1. Similar results were observed for StrategyQA (76.75 → 74.85).

What gave the best results—for both the *Sampling* methods across two datasets—was *Conditional Resampling* with a *different* method from the originally chosen one. It results in a large gain over *Sampling* when the original *Sampling* used subquestion decomposition and *Conditional Resampling* used CoT (75.6 → 76.3) and vice versa (76.4 → 76.8) for GSM8K. This shows that it is useful to change methods using *Conditional Resampling*, a

novel finding with our framework. Furthermore, the results on both datasets on and both the dev and test set show how heterogeneous sampling is complementary to LLM-based selection.

**Resampling without external tools is often insufficient.** We note that the results for *Conditional Resampling* on StrategyQA in Table 1 is more mixed, as the switch from CoT to Subques only results in an insignificant 0.10 change in accuracy (76.75 → 76.85). We hypothesize that because StrategyQA requires factual knowledge, drawing more samples, regardless of the reasoning strategy used, will always be limited by the LLM’s internal knowledge. A real example at the bottom of Figure 4 shows how resampling can preserve an incorrect model-generated claim.

We note that this limitation can be overcome by external tools.<sup>4</sup> het<sub>facts</sub> shows a +2-point improvement (76.75 → 78.75) over *Sampling* for CoT and around +4 points for Subq (76.76 → 80.25) (and *Selection* further improves accuracy).

## 4.3 SCREWS generalizes across LLMs

Table 1 shows the accuracy after each module of SCREWS. As discussed above, a mixture of strategies followed by *Selection* performs best. We then evaluated this strategy on the test set for each task across three LLMs and against a prior work using Self-Refine (Madaan et al., 2023). We report the results in Table 2.

Self-Refine gets a gain of +0.2 points on GSM8K dataset using both ChatGPT and GPT-4 models, our heterogeneous resampling (het<sub>res</sub>) achieves a gain

<sup>4</sup>Recall we are only including facts in *Conditional Resampling*. In preliminary studies we found that including facts during *Sampling* instead results in accuracy above 90%.

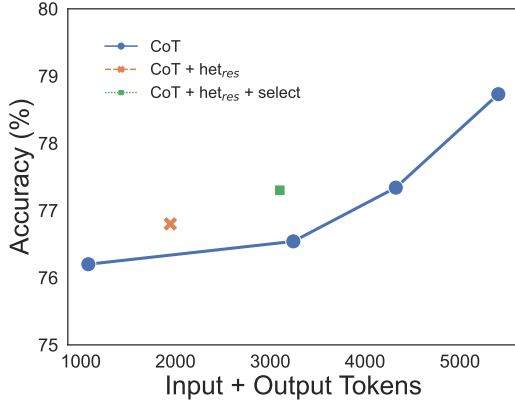


Figure 3: On GSM8K, sampling cost vs. accuracy. The blue line shows a baseline of majority voting over  $k \in \{1, 3, 4, 5\}$  CoT samples. The shaped points are the our proposed strategies from Sec. 4 that use CoT with  $het_{res}$  and selection (select).

of around +0.5 points with chain of thought strategy. Moreover, combining it with selection leads to a gain of +1.1 points for ChatGPT (76.20  $\rightarrow$  77.30) and +1.7 for GPT-4 (92.18  $\rightarrow$  93.88). On Llama 2 70B, an open-source LLM independent of OpenAI APIs, we achieve a gain of +0.6 points (59.24  $\rightarrow$  59.88). Note that self-refine resamples every sample, while our conditional resampling is needed for a small fraction of the inputs, proving it to be a more cost effective and accurate strategy. The results follows a similar trend where selection with heterogeneous sampling leads to an improvement of +1.7 points for Llama 2 70B (74.1  $\rightarrow$  75.8) and +2.2 points for ChatGPT (75.7  $\rightarrow$  77.9). We do not report StrategyQA scores with GPT-4 as even the baseline achieves nearly perfect accuracy.

The results follows a similar trend for subquestion decomposition (Table 2, right). For GSM8K, selection over heterogeneous resampling leads to a gain of +1 point for Llama 2 70B (55.66  $\rightarrow$  56.64), +1.2 points for ChatGPT (77.1  $\rightarrow$  78.3) and +2.2 points for GPT-4 (91.24  $\rightarrow$  93.44). For StrategyQA, the results are improved by +2.4 points for Llama 2 70B (73.25  $\rightarrow$  75.65) and +2.1 points for ChatGPT (76.10  $\rightarrow$  78.25).

## 5 Analysis and Ablations

### 5.1 Total Cost

SCREWS supports many methods with different cost/accuracy tradeoffs. Notably, our methodology of conditional resampling and selection incurs additional LLM calls. In this analysis, we compare against the simpler method of  $k$ -shot sampling with a single sampling strategy and selecting via a ma-

jority vote. We evaluate  $k = \{1, 3, 4, 5\}$  chain of thought (CoT) samples for the test set of GSM8K with ChatGPT ( $\bullet$ ). The first sample is generated at temp = 0 and the remaining use temp = 0.7, and ties are broken randomly. Fig. 3 displays the strategies that use CoT with our proposed approach of heterogeneous resampling and selection on GSM8K. The cost is represented as the total count of input tokens (prompt + query) and output tokens for all LLM calls needed by that strategy, averaged over test examples. Generally, subquestion decomposition is expensive as it is costly to call  $\psi_{question}$ . However, it is affordable to use it in *Conditional Resampling* ( $\times$ ), since resampling only occurs a fraction of the time. This method is both cheaper and more accurate than *Sampling* majority voted 3 samples with CoT. Combining it with selection ( $\blacksquare$ ) achieves an accuracy better than 4 CoT samples majority voted at roughly the cost of 3 samples. Appendix B discusses a detailed breakdown of each module’s input and output token costs.

### 5.2 Oracle Subquestions

Recall that the GSM8K dataset includes subquestions that were generated by their own fine-tuned model with some access to the solution. We will refer to these as *oracle subquestions*.

Replacing our model-generated questions with oracle subquestions unsurprisingly improves the baseline subquestion performance on the test set (76.10  $\rightarrow$  78.60, ChatGPT). Still, we find that heterogeneous sampling (79.10) and selection (79.45) results in further improvement. This reaffirms the importance of heterogenous sampling and selection, and that stronger reasoning strategies (oracle) can benefit from relatively weaker ones (CoT).

### 5.3 More Revision Steps

Section 4 showed that on GSM8K (test) with ChatGPT, revising with heterogeneous resampling improves the CoT strategy (76.2  $\rightarrow$  76.8) and selection improves it further (77.3). Madaan et al. (2023) find that the score increase after each iteration of SR is marginal. We likewise test additional iterations of *Conditional Resampling+Selection*, resulting in scores of [77.6, 77.9, 77.8, 77.6] on iterations 2-5 respectively. We conclude that these improvements are also relatively minor compared to first iteration of *Conditional Resampling*. However, like those of Madaan et al. (2023), these gains are not negligible, and it may be helpful to run some further iterations, cost permitting.

---

**Question** In a family of 5, three people eat three eggs each day while the rest eat two eggs each day. If they eat eggs every day, how many eggs will the family consume in a week?

**Sampling (A)** Three persons eat  $3 \times 3 = \langle 3 \times 3 = 9 \rangle 9$  eggs each day. So,  $5 - 3 = \langle 5 - 3 = 2 \rangle 2$  persons eat two eggs in a day. Thus,  $2 \times 2 = 4$  eggs are consumed by the two other people in a day. So the family consumes  $9 + 4 = \langle 9 + 4 = 13 \rangle 13$  eggs each day. Therefore,  $13 \times 7 = \langle 13 \times 7 = 91 \rangle 91$  eggs will be consumed by the family in a week. The answer is 91. ✓

**Conditional Resampling (B)** The three people eat 3 eggs each day, so they eat a total of  $3 \times 3 = \langle 3 \times 3 = 9 \rangle 9$  eggs each day. The two people eat 2 eggs each day, so they eat a total of  $2 \times 2 = \langle 2 \times 2 = 4 \rangle 4$  eggs each day. The remaining two people eat a total of 4 eggs each day - 3 eggs eaten by the other three people =  $\langle 4 - 3 = 1 \rangle 1$  egg each day. The family consumes a total of 9 eggs + 4 eggs + 1 egg =  $\langle 9 + 4 + 1 = 14 \rangle 14$  eggs each day. Therefore, the family will consume 14 eggs/day  $\times 7$  days/week =  $\langle 14 \times 7 = 98 \rangle 98$  eggs in a week. The answer is 98. ✗

**Selection (A)** ✓

---

**Question** The local firefighters are doing a “fill the boot” fundraiser. Their goal is to raise \$6300. After the first 3 hours, they have raised \$2100. For how many hours do they have to fundraise in total to reach their goal, assuming an equal amount raised in every hour?

**Sampling (A)** They need to raise  $\$6300 - \$2100 = \langle \$6300 - 2100 = 4200 \rangle 4200$  more dollars. If they raise an equal amount in every hour, they need to fundraise for  $\$4200 / 1 = \langle \$4200 / 1 = 4200 \rangle 4200$  hours. The answer is 4200. ✗

**Conditional Resampling (B)** The firemen raised \$2100 in 3 hours, so they raised  $\$2100 / 3$  hours =  $\langle \$2100 / 3 = 700 \rangle 700$  per hour. To reach their goal of \$6300, they need to fundraise for  $\$6300 / \$700$  per hour =  $\langle 6300 / 700 = 9 \rangle 9$  hours. The answer is 9. ✓

**Selection (B)** ✓

---

**Question:** Would Alexander Hamilton have known about koalas?

**Sample:** Alexander Hamilton died in 1804. The first koala specimen was collected in 1792. It is possible that Hamilton knew about koalas, but it is unlikely. The answer is Unknown ✗

**Conditional Resampling:** Alexander Hamilton died in 1804. The first koala specimen was collected in 1792. It is possible that Hamilton knew about koalas. The answer is True ✗

**Facts:** Alexander Hamilton died in 1804. The first published depiction of a koala was in 1810

**Tool-based Resample:** No, Alexander Hamilton would not have known about koalas. The answer is False ✓

---

Figure 4: The top two examples demonstrate the importance of the *Selection* module for the GSM8K dataset. The last example shows how tool use (“Facts”) can be helpful for the StrategyQA dataset.

## 5.4 Model-based selection vs. majority voting

As discussed in Section 2.4, majority voting across  $k$  samples is a common method for *Selection*. Typically, this involves sampling  $k$  answers from  $\psi$  at high temperature, as we did in Section 5.1. We now consider majority voting when samples are drawn from *heterogeneous* strategies. In particular, for each question in GSM8K, we use ChatGPT to generate three responses using CoT and subquestion decomposition from Table 2 and oracle subquestions from Section 5.2. We compare majority voting with a model-based selector,  $\psi_{\text{select}}$ , that includes all 3 responses in the prompt. We find that model based selector (86.90) outperforms majority voting (85.52) by +1.4 points; yet much lower than the upper bound of 92.50 (with a perfect selector).

## 5.5 Selected Examples

The top two examples of Figure 4, on the GSM8K dataset, demonstrate the usefulness of the *Selection* module. The first example shows how an error introduced by *Conditional Resampling* can be reverted by *Selection*. The second example shows how a correction found by *Conditional Resampling* can be kept by *Selection*. The last example in Figure 4, on the StrategyQA dataset, illustrates that ordinary *Resampling* is unlikely to correct an incorrect fact generated by the LLM. However, providing the correct facts during *Resampling* gives the model access to new information, leading to the

correct answer.

## 6 Conclusion and Future Work

We have proposed SCREWS, a modular reasoning-with-revisions framework to answer reasoning questions with LLMs. Based on our experiments we draw the two primarily conclusions: 1) **Selection plays an important role:** Although *Conditional Resampling* often improves the result of *Sampling*, *Selection* can help avoid errors from the case where it does not. It was beneficial on all three datasets; and 2) **Heterogeneous vs. homogeneous resampling:** Using different reasoning methods for *Sampling* and *Conditional Resampling* can lead to higher accuracy, with or without *Selection*. Furthermore, we demonstrate the effectiveness of SCREWS as a framework for determining useful and cost-effective strategies.

SCREWS combines the three important modules *Sampling*, *Conditional Resampling* and *Selection* in a modular framework. While the relative ordering of reasoning strategies appears robust within a single dataset, the best configuration of modules may vary by task and could be identified through a method such as exhaustive search, Monte Carlo Tree Search, or reinforcement learning. Further, the modules themselves could be fine-tuned to improve end-to-end performance. We leave this for future work and encourage a broader spectrum of strategies for reasoning with LLMs.



## 7 Limitations

Our work explores a framework for revision that we experimented on two reasoning datasets: arithmetic reasoning and fact-based question answering. Although our framework can be extended to other tasks due to the flexible nature of our framework, we did not test it on other tasks. We leave this for future work. Also, we explored two most popular and effective reasoning *Sampling* strategies of chain of thought and subquestion decomposition, so many other methods including but not limited to PAL (Gao et al., 2022) and Faithful CoT (Lyu et al., 2023) were not tested. To investigate our research questions, we permitted use of oracle or gold auxiliary labels (subquestions and facts); thus, our best numbers are not intended to be treated as comparable to SOTA.

Due to the ever-changing nature of OpenAI’s closed-source APIs, some results may not be reproducible in the future. However, we are releasing the prompts and code for our work and included a set of results using the open source Llama 2 70B model. Finally, the output of LLMs can be sensitive to changes in input prompt phrasing. Minor changes in the prompt can lead to different responses, suggesting that the models may not consistently apply the same reasoning or context.

## 8 Ethical Considerations

The authors recognize that any work in advancing reasoning strategies using LLMs can be used to advance capabilities on malicious tasks. Besides this potential for dual use, the authors do not see additional ethical concerns.

## References

- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *ArXiv*, abs/2204.05862. 618-630
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc. 631-644
- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. [Learning to rank using gradient descent](#). In *Proceedings of the 22nd International Conference on Machine Learning, ICML ’05*, page 89–96, New York, NY, USA. Association for Computing Machinery. 645-651
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. [Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks](#). *ArXiv*, abs/2211.12588. 652-656
- Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. [Teaching large language models to self-debug](#). *ArXiv*, abs/2304.05128. 657-659
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. [Deep reinforcement learning from human preferences](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 660-663
- 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](#). *ArXiv*, abs/2110.14168. 664-670
- Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. [LM vs LM: Detecting factual errors via cross examination](#). *ArXiv*, abs/2305.13281. 671-673

674	Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. <a href="#">NL-EDIT: Correcting semantic parse errors through natural language interaction</a> . In <i>Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 5599–5610, Online. Association for Computational Linguistics.	731
675		732
676		733
677		734
678		735
679		736
680		737
681		
682		
683	Jinlan Fu, See-Kiong Ng, and Zhengbao Jiang and Pengfei Liu. 2023. <a href="#">GPTScore: Evaluate as you desire</a> . <i>ArXiv</i> , abs/2302.04166.	738
684		739
685		740
686	Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. <a href="#">Pal: Program-aided language models</a> . <i>ArXiv</i> , abs/2211.10435.	741
687		742
688		743
689		744
690	Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. <a href="#">Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies</a> . <i>Transactions of the Association for Computational Linguistics</i> , 9:346–361.	745
691		746
692		747
693		748
694		749
695		750
696		751
697		
698		
699		
700	Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. <a href="#">Large language models can self-improve</a> . <i>ArXiv</i> , abs/2210.11610.	752
701		753
702		754
703		755
704		
705	Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. <a href="#">Large language models cannot self-correct reasoning yet</a> . <i>ArXiv</i> , abs/2310.01798.	756
706		757
707		758
708		759
709		760
710		761
711		762
712		763
713		764
714		765
715		766
716		767
717		768
718		769
719		770
720		771
721		772
722		773
723		774
724		775
725		776
726		777
727		778
728		779
729		780
730		781
		782
		783
		784
		785
		786
		787
		788
		789
		790
		791
		792
		793
		794
		795
		796
		797
		798
		799
		800
		801
		802
		803
		804
		805
		806
		807
		808
		809
		810
		811
		812
		813
		814
		815
		816
		817
		818
		819
		820
		821
		822
		823
		824
		825
		826
		827
		828
		829
		830
		831
		832
		833
		834
		835
		836
		837
		838
		839
		840
		841
		842
		843
		844
		845
		846
		847
		848
		849
		850
		851
		852
		853
		854
		855
		856
		857
		858
		859
		860
		861
		862
		863
		864
		865
		866
		867
		868
		869
		870
		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
		918
		919
		920
		921
		922
		923
		924
		925
		926
		927
		928
		929
		930
		931
		932
		933
		934
		935
		936
		937
		938
		939
		940
		941
		942
		943
		944
		945
		946
		947
		948
		949
		950
		951
		952
		953
		954
		955
		956
		957
		958
		959
		960
		961
		962
		963
		964
		965
		966
		967
		968
		969
		970
		971
		972
		973
		974
		975
		976
		977
		978
		979
		980
		981
		982
		983
		984
		985
		986
		987
		988
		989
		990
		991
		992
		993
		994
		995
		996
		997
		998
		999
		1000

787	<a href="#">your facts and try again: Improving large language models with external knowledge and automated feedback.</a> <i>ArXiv</i> , abs/2302.12813.		
788			
789			
790	Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson E. Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, John Kernion, Kamil.e Lukovsiute, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkat Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Janina Brauner, Sam Bowman, and Ethan Perez. 2023. <a href="#">Question decomposition improves the faithfulness of model-generated reasoning.</a> <i>ArXiv</i> , abs/2307.11768.		
791			
792			
793			
794			
795			
796			
797			
798			
799			
800			
801	Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. <a href="#">Toolformer: Language models can teach themselves to use tools.</a> <i>ArXiv</i> , abs/2302.04761.		
802			
803			
804			
805			
806	Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. <a href="#">Reflexion: Language agents with verbal reinforcement learning.</a>		
807			
808			
809			
810	Kumar Shridhar, Jakob Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. <a href="#">Automatic generation of socratic subquestions for teaching math word problems.</a> In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.		
811			
812			
813			
814			
815			
816			
817			
818	Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. <a href="#">Distilling reasoning capabilities into smaller language models.</a> In <i>Findings of the Association for Computational Linguistics: ACL 2023</i> , pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.		
819			
820			
821			
822			
823			
824	Niket Tandon, Aman Madaan, Peter Clark, Keisuke Sakaguchi, and Yiming Yang. 2021. <a href="#">Interscript: A dataset for interactive learning of scripts through error feedback.</a> <i>ArXiv</i> , abs/2112.07867.		
825			
826			
827			
828	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		
829			
830			
831			
832			
833			
834			
835			
836			
837			
838			
839			
840			
841			
842			
843			
844			
		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 Thomas Scialom. 2023. <a href="#">Llama 2: Open foundation and fine-tuned chat models.</a> <i>ArXiv</i> , abs/2307.09288.	845
			846
			847
			848
			849
			850
		Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022. <a href="#">Self-Consistency improves chain of thought reasoning in language models.</a> <i>ArXiv</i> , abs/2203.11171.	851
			852
			853
			854
		Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. <a href="#">Chain-of-thought prompting elicits reasoning in large language models.</a> In <i>Advances in Neural Information Processing Systems</i> , volume 35, page 24824–24837. Curran Associates, Inc.	855
			856
			857
			858
			859
			860
			861
		Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. <a href="#">Generating sequences by learning to self-correct.</a> <i>ArXiv</i> , abs/2211.00053.	862
			863
			864
			865
		Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. 2022. <a href="#">Large language models are better reasoners with self-verification.</a> volume abs/2212.09561.	866
			867
			868
			869
		Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. <a href="#">Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in llms.</a> <i>ArXiv</i> , abs/2306.13063.	870
			871
			872
			873
			874
		Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. <a href="#">Re3: Generating longer stories with recursive reprompting and revision.</a> In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	875
			876
			877
			878
			879
			880
			881
		Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. <a href="#">Tree of thoughts: Deliberate problem solving with large language models.</a> <i>ArXiv</i> , abs/2305.10601.	882
			883
			884
			885
			886
		Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. <a href="#">ReAct: Synergizing reasoning and acting in language models.</a> <i>ArXiv</i> , abs/2210.03629.	887
			888
			889
			890
		Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. 2022. <a href="#">Automatic chain of thought prompting in large language models.</a> <i>ArXiv</i> , abs/2210.03493.	891
			892
			893
			894
		Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. <a href="#">Progressive-Hint Prompting improves reasoning in large language models.</a> <i>ArXiv</i> , abs/2304.09797.	895
			896
			897
			898

899	Denny Zhou, Nathanael Scharli, Le Hou, Jason Wei,	<b>A Detailed comparison with related work</b>	910
900	Nathan Scales, Xuezhi Wang, Dale Schuurmans,	In <a href="#">Figure 5</a> , we contrast the framework of	911
901	Olivier Bousquet, Quoc Le, and Ed Huai hsin	SCREWS as described in <a href="#">Figure 2</a> with other re-	912
902	Chi. 2022. <a href="#">Least-to-most prompting enables com-</a>	cent work on reasoning via iterations.	913
903	<a href="#">plex reasoning in large language models.</a> <i>ArXiv,</i>		
904	<a href="#">abs/2205.10625.</a>		
905	Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B.	<b>B Token Cost</b>	914
906	Brown, Alec Radford, Dario Amodei, Paul Chris-	Table 3 shows the token cost of input and output	915
907	tiano, and Geoffrey Irving. 2019. <a href="#">Fine-tuning lan-</a>	for each module in SCREWS. Due to its iterative	916
908	<a href="#">guage models from human preferences.</a> <i>ArXiv,</i>	nature, subquestion decomposition requires on av-	917
909	<a href="#">abs/1909.08593.</a>	erage four times more input tokens than the other	918
		modules. For <i>Conditional Resampling</i> , the model	919
		first predicts whether it wants to modify its output	920
		or not, using one token (“Yes” or “No”) for each	921
		sample and then only for the answers starting with	922
		“No”, it resamples. For the <i>Selection</i> module, the	923
		model chooses one of the two samples presented to	924
		it, using one token (A or B) for the output. <a href="#">Table 4</a>	925
		directly reports the input and output token cost of	926
		our proposed strategies from <a href="#">Table 2</a> .	927
		<b>C Prompts</b>	928
		Below are abbreviated versions of the prompts	929
		used in the experiments, including instructions and	930
		demonstrations. For readability, we show only 1–2	931
		demonstrations in each prompt. In each demonstra-	932
		tion, the demonstrated result string is <b>highlighted</b>	933
		for the reader’s convenience, but this highlighting	934
		is not included in the prompt. Each prompt shown	935
		would be followed by the test question and then	936
		the cue (e.g., “Answer:”) that indicates that a result	937
		string should follow.	938
		<b>C.1 Sampling</b>	939
		For Chain of Thought (CoT) and Subquestion De-	940
		composition for GSM8K and StrategyQA, 5-shot	941
		prompts were used.	942
		<b>C.1.1 Chain of Thought</b>	943
		<b>GSM8K</b>	944
		I am a highly intelligent question answering bot. I will answer	945
		the last question ‘Question’ providing equation in « » format	946
		in step by step manner.	947
		Question: James writes a 3-page letter to 2 different	948
		friends twice a week. How many pages does he write a year?	949
		Answer: <b>He writes each friend <math>3 * 2 = «3 * 2 = 6»6</math> pages</b>	950
		<b>a week. So he writes <math>6 * 2 = «6 * 2 = 12»12</math> pages every</b>	951
		<b>week. That means he writes <math>12 * 52 = «12 * 52 = 624»624</math></b>	952
		<b>pages a year. The answer is 624</b>	953
		<hr/>	954
		<b>StrategyQA</b>	955
		You are a highly intelligent question answering bot. You will	956
		answer the question ‘Question’ in as details as possible.	957

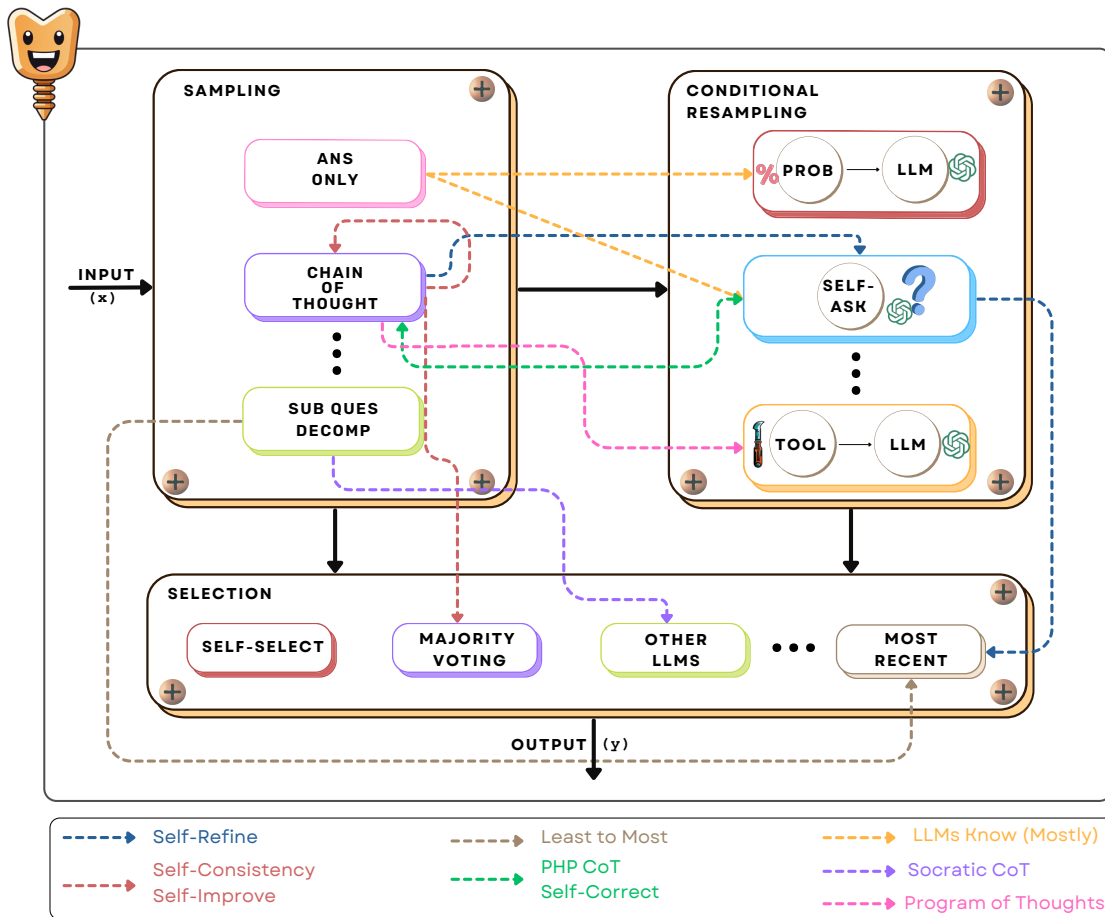


Figure 5: Overview of our modular framework for reasoning with revisions, SCREWS. Each of the three large boxes (“modules”) contains several alternatives (“submodules”). Several past works can be instantiated using our framework and can be combined with the heterogeneous resampling and selection strategy. Some examples are presented, namely Self-Refine (Madaan et al., 2023), Least to Most (Zhou et al., 2022), LLMs Know (Mostly) (Kadavath et al., 2022), Self-Consistency (Wang et al., 2022), Self-Improve (Huang et al., 2022), PHP CoT (Zheng et al., 2023), Self-Correct (Welleck et al., 2022), Socratic CoT (Shridhar et al., 2022), Program of Thoughts (Chen et al., 2022). (...) represents other sub-components that can be added to each module, like cached memory or web search for *Sampling*, among others.

958 Question: Is coal needed to practice parachuting?  
 959 Answer: Parachuting requires a parachute. Parachutes are  
 960 made from nylon. Nylon is made from coal. The answer is  
 961 True  
 962

### 963 C.1.2 Subquestion Decomposition

964 While subquestion decomposition uses a single  
 965 prompt, each example requires multiple API calls  
 966 because the next subquestion needs to be appended  
 967 to the prompt.

#### 968 GSM8K

969 I am a highly intelligent question answering bot. I will answer  
 970 the last question ‘Q’ providing equation in « » format keeping  
 971 the Problem and previous Q and A into account.

972 Problem: There are 5 houses on a street, and each of the  
 973 first four houses has 3 gnomes in the garden. If there are a  
 974 total of 20 gnomes on the street, how many gnomes does the

975 fifth house have?

976 Q: How many gnomes are in the first four houses?

977 A: In the first four houses, there are a total of 4 houses \* 3  
 978 gnomes = «4 \* 3 = 12»12 gnomes. The answer is 12

979 Q: How many gnomes does the fifth house have?

980 A: Therefore, the fifth house had 20 total gnomes - 12 gnomes  
 981 = «20 - 12 = 8»8 gnomes. The answer is 8

#### 982 StrategyQA

983 You are a highly intelligent question answering bot. You will  
 984 answer the last question ‘Q’ keeping the Problem and previous  
 985 Q and A into account and then answer the Final Question  
 986 based on all the previous answer ‘A’.  
 987

988 Problem: Is coal needed to practice parachuting?

989 Q: What is one of the most important item that you need to go  
 990 parachuting?

991 A: Parachuting requires a parachute.

992 Q: What is #1 made out of?

Method	Input Tokens	Output Tokens	Total Tokens
<b>Subquestion generation step <math>\psi_{\text{question}}</math></b>			
Subques	360	180	540
<b>Sampling step <math>\psi</math></b>			
CoT	774	307	1081
CoT ( $k = 5$ )	3870	1530	5400
Subques	3121	434	3555
Subques (Oracle)	3187	413	3600
<b>Conditional Resampling step <math>\psi_{\text{ask}}</math></b>			
CoT	869	105	1184
Subques	3780	136	3916
Subques (Oracle)	3525	131	3656
<b>Selection step <math>\psi_{\text{select}}</math></b>			
Selection	1296	1	1297

Table 3: Average counts of input and output tokens for each choice of each module (step) in SCREWS. Many of the methods in Tab. 2 need to call multiple modules. We remark that the input tokens at each step include output tokens from previous steps. The counts shown for later steps average not only over examples, but also over choices of method for the previous steps.

Cost	Method	Chain-of-thought			Subquestion Decomposition		
		Llama2-70B	ChatGPT	GPT-4	Llama2-70B	ChatGPT	GPT-4
Input Token	Baseline	774	774	774	3121	3121	3121
	+ hetres	780	612	624	217	231	194
	+ select	324	259	262	319	324	276
Output Token	Baseline	362	307	292	454	434	429
	+ hetres	35	28	24	34	31	30
	+ select	1	1	1	1	1	1

Table 4: This companion table to Table 2 reports the exact input and output token costs, averaged per example across the full test set for GSM8K.

993 A: Parachutes are made from nylon.  
994 Q: Is #2 originally made from coal?  
995 A: Nylon is made from coal.  
996 Final Question: Is coal needed to practice parachuting?  
997 Final Answer: True

## 999 C.2 Conditional Resampling

1000 We used 4-shot prompting for *conditional resampling*  
1001 (two positive and two negative) for GSM8K  
1002 and StrategyQA. One pair of each is presented be-  
1003 low:

### 1004 C.2.1 Chain of Thought

#### 1005 GSM8K

1006 Question: Natalia sold clips to 48 of her friends in April, and  
1007 then she sold half as many clips in May. How many clips did  
1008 Natalia sell altogether in April and May?

1009 Answer: In April, Natalia sold 48 clips. In May, Natalia sold  
1010 half of the total clips meaning  $72/2 = 36$  clips. So,  
1011 Natalia sold a total of  $48 + 36 = 84$  clips in  
1012 April and May. The answer is 84

1013 You are a math teacher. Do you think the reasoning process  
1014 for the given problem is correct? Let’s check the ‘Answer’

1015 in details, and then decide ‘Yes’ or ‘No’ and then write the  
1016 correct ‘Final Answer’.

1017 Answer: No, the reasoning is incorrect. Final Answer: Natalia  
1018 sold  $48/2 = 24$  clips in May. So she sold a total  
1019 of  $48 + 24 = 72$  clips. The answer is 72

1020 Question: There are 5 houses on a street, and each of the  
1021 first four houses has 3 gnomes in the garden. If there are a  
1022 total of 20 gnomes on the street, how many gnomes does the  
1023 fifth house have?

1024 Answer: In the first four houses, there are a total of 4 houses \*  
1025 3 gnomes = 12 gnomes. Therefore, the fifth  
1026 house had 20 total gnomes - 12 gnomes = 8 gnomes.  
1027 The answer is 8

1028 You are a math teacher. Do you think the reasoning process  
1029 for the given problem is correct? Let’s check the ‘Answer’  
1030 in details, and then decide ‘Yes’ or ‘No’ and then write the  
1031 correct ‘Final Answer’.

1032 Answer: Yes, the reasoning is correct. Final Answer: In the  
1033 first four houses, there are a total of 4 houses \* 3 gnomes  
1034 = 12 gnomes. Therefore, the fifth house had 20  
1035 total gnomes - 12 gnomes = 8 gnomes. The  
1036 answer is 8

---

### StrategyQA

Question: Could Durian cause someone's stomach to feel unwell?

Answer: Durian has a pungent odor that many people describe as being similar to feet and onions. Unpleasant smells can make people feel excited and they like it. The answer is False  
You are an expert teacher. Do you think the reasoning process for the given problem is correct? Let's check the 'Answer' in details, and then decide 'Yes' or 'No' and then write the correct 'Final Answer'.

Answer: **No, the reasoning is incorrect. Final Answer: Durian has a pungent odor that many people describe as being similar to feet and onions. Unpleasant smells can make people feel nauseous. The answer is True**

Question: Was Daniel thrown into the lion's den in the New Testament?

Answer: The Book of Daniel is a book in the Old Testament of the Bible. The Bible is divided into the Old Testament and the New Testament. The New Testament focuses on four Gospels regarding the life of Jesus. The answer is False

You are an expert teacher. Do you think the reasoning process for the given problem is correct? Let's check the 'Answer' in details, and then decide 'Yes' or 'No' and then write the correct 'Final Answer'.

Answer: **Yes, the reasoning is correct. Final Answer: The Book of Daniel is a book in the Old Testament of the Bible. The Bible is divided into the Old Testament and the New Testament. The New Testament focuses on four Gospels regarding the life of Jesus. The answer is False**

---

### StrategyQA (Resampling with facts)

You are a highly intelligent question answering bot. You will answer the question 'Question' in as details as possible. 'Facts' are provided to assist you in answering the questions.  
Question: Are vinegar pickled cucumbers rich in lactobacillus?

Facts: Pickles made with vinegar are not probiotic and are simply preserved. Pickles made through a soak in a salt brine solution begin to ferment because of lactobacillus.

Answer: **No, vinegar does not contain lactobacillus. The answer is False**

Question: Does Masaharu Morimoto rely on glutamic acid?

Facts: Masaharu Morimoto is a Japanese chef. Japanese cuisine relies on several forms of seaweed as ingredients and flavorings for broth like kombu dashi. Glutamic acid has been identified as the flavoring component in kombu seaweed.

Answer: **Yes, Japanese chef uses a lot of glutamic acid. The answer is True**

---

## C.2.2 Subquestion Decomposition GSM8K

Here is a math question and its solution.

Problem: Noah is a painter. He paints pictures and sells them at the park. He charges \$60 for a large painting and \$30 for a small painting. Last month he sold eight large paintings and

four small paintings. If he sold twice as much this month, how much is his sales for this month?

How much did Noah earn from the large paintings? Noah earned \$60/large painting x 8 large paintings = \$«60 \* 8 = 480»480 for the large paintings. The answer is 480

Question: How much did Noah earn from the small paintings?  
Answer: He also earned \$60/small painting x 4 small paintings = \$«60 \* 4 = 240»240 for the small paintings. The answer is 240

You are a math teacher. Do you think the reasoning process for the given problem is correct? Let's check the 'Answer' in details, and then decide 'Yes' or 'No' and then write the correct 'Final Answer'.

Answer: **No, the reasoning is incorrect. Final Answer: He also earned \$30/small painting x 4 small paintings = \$«30 \* 4 = 120»120 for the small paintings. The answer is 120**

Here is a math question and its solution.

Problem: To make pizza, together with other ingredients, Kimber needs 10 cups of water, 16 cups of flour, and 1/2 times as many teaspoons of salt as the number of cups of flour. Calculate the combined total number of cups of water, flour, and teaspoons of salt that she needs to make the pizza.

How many teaspoons of salt does Kimber need? To make the pizza, Kimber half as many teaspoons of salt as the number of cups of flour, meaning she needs  $1/2 * 16 = «16 * 1/2 = 8»8$  teaspoons of salt. The answer is 8

How many cups of flour and teaspoons of salt does Kimber need? The total number of cups of flour and teaspoons of salt she needs is  $8 + 16 = «8 + 16 = 24»24$ . The answer is 24

Question: How many cups of water, flour, and salt does Kimber need?

Answer: She also needs 10 cups of water, which means the total number of cups of water and flour and teaspoons of salt she needs is  $24 + 10 = «24 + 10 = 34»34$ . The answer is 34

You are a math teacher. Do you think the reasoning process for the given problem is correct? Let's check the 'Answer' in details, and then decide 'Yes' or 'No' and then write the correct 'Final Answer'.

Answer: **Yes, the reasoning is correct. Final Answer: She also needs 10 cups of water, which means the total number of cups of water and flour and teaspoons of salt she needs is  $24 + 10 = «24 + 10 = 34 > »34$ . The answer is 34**

---

### StrategyQA

Here is a question and its answer.

Context: Would a diet of ice eventually kill a person?

Ice is the solid state of what? Ice can be melted into water, which consists of hydrogen and oxygen.

What nutrients are needed to sustain human life? Humans need carbohydrates, proteins, and fats that are contained in foods.

Question: Are most of #2 absent from #1?

Answer: Water does not contain fat, carbohydrates or protein. You are an expert teacher. Based on the provided context, do you think the reasoning process for the given problem is correct? Let's check the 'Answer' in details, and then decide 'Yes' or 'No' and then write the correct 'Final Answer'.

Answer: **Yes, the reasoning is correct. Final Answer: Water does not contain fat, carbohydrates or protein.**

1157  
 1158 Here is a question and its answer.  
 1159 Context: Can binary numbers and standard alphabet satisfy  
 1160 criteria for a strong password?  
 1161 Which characters make up binary numbers? Binary numbers  
 1162 only contain 0 and 1.  
 1163 Which characters make up the standard English alphabet? The  
 1164 standard alphabet contains twenty six letters but no special  
 1165 characters.  
 1166 Question: Does #1 or #2 include special characters or  
 1167 symbols?  
 1168 Answer: Yes, it contains all the special characters.  
 1169 You are an expert teacher. Based on the provided context,  
 1170 do you think the reasoning process for the given problem is  
 1171 correct? Let's check the 'Answer' in details, and then decide  
 1172 'Yes' or 'No' and then write the correct 'Final Answer'.  
 1173 Answer: **No, the reasoning is incorrect. Final Answer: Neither**  
 1174 **binary digits nor English alphabets consists of any special**  
 1175 **characters which is needed for a strong password.**

### 1177 C.3 Selection

1178 The LLM-based selection module  $\psi_{\text{select}}$  uses a 2-  
 1179 shot prompt. The 2 demonstrations in the prompt  
 1180 are shown below, for each dataset.

#### 1181 GSM8K

1182 You are an expert math teacher. You are provided with a  
 1183 question and two answers. Lets check the 'Answer choices'  
 1184 step by step, and then decide which answer is correct '(A)' or  
 1185 '(B)'

1186 Question: Natalia sold clips to 48 of her friends in April, and  
 1187 then she sold half as many clips in May. How many clips did  
 1188 Natalia sell altogether in April and May?

1189 Answer choices:

1190 (A) In April, Natalia sold 48 clips. In May, Natalia sold 24  
 1191 clips. So, Natalia sold a total of 72 clips in April and May.  
 1192 The answer is 72. So in May she sold 48 clips. Total clips  
 1193 sold in April and May =  $72 + 48 = \llcorner 72 + 48 = 120 \gg 120$ .  
 1194 The answer is 120

1195 (B) Natalia sold  $48/2 = \llcorner 48/2 = 24 \gg 24$  clips in May. The  
 1196 answer is 24. Natalia sold  $48 + 24 = \llcorner 48 + 24 = 72 \gg 72$  clips  
 1197 altogether. The answer is 72

1198 Answer: **(B)**  
 1199

1200 You are an expert math teacher. You are provided with a  
 1201 question and two answers. Lets check the 'Answer choices'  
 1202 step by step, and then decide which answer is correct '(A)' or  
 1203 '(B)'

1204 Question: Dolly has two books. Pandora has one. If both  
 1205 Dolly and Pandora read each others' books as well as their  
 1206 own, how many books will they collectively read by the end?  
 1207 Answer choices:

1208 (A) There are a total of  $2 + 1 = \llcorner 2 + 1 = 3 \gg 3$  books. The  
 1209 answer is 3. Dolly and Pandora both read all 3 books, so 3  
 1210 books/person  $\times$  2 people =  $\llcorner 3 * 2 = 6 \gg 6$  books total. The  
 1211 answer is 6

1212 (B) The total number of books are  $2 * 1 = \llcorner 2 * 1 = 2 \gg 2$

books. The answer is 2. Dolly and Pandora read each other's  
 books as well as their own, so the total number of books they  
 read is 3 books. The answer is 3

Answer: **(A)**

### StrategyQA

You are the expert in the field. You are provided with a  
 question and two answers. Lets check the reasoning process  
 of each of the answer step by step, and then decide which  
 answer is correct '(A)' or '(B)'

Question: Could Durian cause someone's stomach to feel  
 unwell?

Answer choices:

(A) Durian has a pungent odor that many people describe as  
 being similar to feet and onions. Unpleasant smells can make  
 people feel nauseous. The answer is True

(B) Durian has a pungent odor that many people describe as  
 being similar to feet and onions. Unpleasant smells can make  
 people feel excited and they like it. The answer is False

Answer: **(A)**

You are the expert in the field. You are provided with a  
 question and two answers. Lets check the reasoning process  
 of each of the answer step by step, and then decide which  
 answer is correct '(A)' or '(B)'

Question: Was Daniel thrown into the lion's den in the New  
 Testament?

Answer choices:

(A) The Book of Daniel is a book in the New Testament of  
 the Bible. The Bible is divided into the Old Testament and the  
 New Testament. The New Testament focuses on the life of  
 Daniel. The answer is True

(B) The Book of Daniel is a book in the Old Testament of the  
 Bible. The Bible is divided into the Old Testament and the  
 New Testament. The New Testament focuses on four Gospels  
 regarding the life of Jesus. The answer is False

Answer: **(B)**

### C.4 Question Generation

5-shot prompts were used for generating subques-  
 tions for GSM8K dataset. An example is provided  
 below:

#### GSM8K

I am a highly intelligent question generation bot. I will take  
 the given question 'Q' and will decompose the main question  
 into all 'subquestions' required to solve the question step by  
 step.

Q: James writes a 3-page letter to 2 different friends twice  
 a week. How many pages does he write a year?

Subquestions: **How many pages does he write each week?**  
**How many pages does he write every week? How many pages**  
**does he write a year?**



1268 **StrategyQA**

1269 I am a highly intelligent question generation bot. I will take  
1270 the given question 'Q' and will decompose the main question  
1271 into all 'subquestions' required to solve the question step by  
1272 step.  
1273

1274 Q: Can you buy Casio products at Petco?

1275 Subquestions: What kind of products does Casio manufac-  
1276 ture? What kind of products does Petco sell? Does #1 overlap  
1277 with #2?  
1278

---