

PRICoT: Principle Retrieval and Injection from Inference Successes and Failures for CoT Improvement

Yudai Yamazaki, Naoto Takeda, Yasutaka Nishimura, Kazushi Ikeda,

KDDI Research Inc.

Fujimino, Japan

{yd-yamazaki, no-takeda, yu-nishimura, kz-ikeda}@kddi.com

Abstract

In-Context Learning (ICL) approaches, such as Zero-Shot and Few-Shot prompting, allow Large Language Models (LLMs) to tackle reasoning tasks without additional fine-tuning. However, Zero-Shot prompting often struggles with more complex tasks, whereas Few-Shot prompting demands considerable manual effort and domain expertise to design effective prompts. Although existing work has attempted to alleviate these issues by extracting reasoning rules from carefully crafted, task-specific representative examples, creating or obtaining such examples can be impractical in real-world scenarios. In this paper, we propose a novel approach that enhances the inference accuracy by injecting reasoning principles extracted from QA data, without relying on representative Few-Shot exemplars. This offers a lightweight yet adaptive way to boost accuracy on complex reasoning tasks, while avoiding manual effort and the high exploration costs typical of prior methods. Experiments on benchmarks show that, using GPT-4o, our method outperforms similarity-based Few-Shot and Zero-Shot prompting methods on challenging benchmarks such as GPQA-diamond, achieving an absolute accuracy improvement of up to 2% in scenarios where carefully crafted Few-Shot examples are unavailable.

1 Introduction

In-Context Learning (ICL) enables Large Language Models (LLMs) to improve their performance on a variety of tasks by simply providing relevant task instructions and examples without retraining or fine-tuning the model’s internal parameters (Dong et al., 2024; Wies et al., 2023; Shin et al., 2022; Wang et al., 2022). Two representative forms of ICL are *Zero-Shot Prompting*, which encourages reasoning based solely on general instructions (e.g., “Let’s think step by step”) such as Zero-Shot Chain of Thought (CoT) (Kojima

et al., 2022), and *Few-Shot Prompting* (Brown et al., 2020), which provides a handful of task-specific examples, sometimes accompanied by their corresponding CoT (Wei et al., 2022; Nachane et al., 2024) to guide the model toward more accurate answers.

Despite the effectiveness of these approaches, several challenges remain. (1) Zero-Shot Prompting often struggles with complex tasks because it heavily relies on the model’s inherent knowledge (Labrak et al., 2024; Shaikh et al., 2023). (2) Few-Shot Prompting typically requires careful design and selection of examples by an expert familiar with the task (Liu et al., 2023; Zhao et al., 2021). Whenever the task changes, practitioners must redesign prompts and examples, incurring effort. Consequently, there is a growing demand for methods that can automatically construct prompts (Li et al., 2025).

Existing methods can be broadly categorized based on the availability of representative Few-Shot examples—each setting introducing its own set of challenges.

When Few-Shot examples are entirely unavailable, methods based purely on Zero-Shot prompting must be relied upon. A notable example is Plan and Solve (Wang et al., 2023), which enhances basic Zero-Shot Chain-of-Thought prompting by first generating a high-level plan before reasoning through the task. While such approaches avoid the need for exemplars, their performance is limited on complex tasks due to the lack of task-specific guidance.

In settings where raw task data is available but representative Few-Shot examples are not identified, several methods—such as APE (Zhou et al., 2022) and DSPy (Khattab et al., 2023)—have been proposed to automatically construct prompts using LLMs. These methods iteratively optimize prompts based on evaluation feedback. While effective, they often involve repeated trial-and-error

Zero-Shot Prompting (Plan and Solve)

Model Input

Question: Two friends are eating a jar full of candies. Had P eaten alone, it would have taken him 10 minutes to finish the candies in the jar. Had Q eaten alone, it would have taken her 5 minutes to finish half the jar. Since both of them are eating simultaneously, how many minutes would it take them to empty the jar?

Options: A. 4, B. 5, C. 6, D. 7, E. 8

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

Model Output

P's rate is 1 jar / 10 minutes = 0.1 [jar/min].
 Q's rate is 1 jar / 5 minutes = 0.1 [jar/min].
 P's rate + Q's rate = 0.1 + 0.2 = 0.3 [jar/min].
 $t = 1 / 0.3 = 3.33$ minutes.
Final Answer: A. 4 ❌

Few-Shot Prompting

Model Input

Question: You and your friend spent a total of \$15 for lunch. Your friend spent \$1 more than you. How much did your friend spend on their lunch?

Options: A. \$9, B. \$8, C. \$4, D. \$6, E. \$5

Answer: B

Question: Two friends are eating a jar full of candies. Had P eaten alone, it would have taken him 10 minutes to finish the candies in the jar. Had Q eaten alone, it would have taken her 5 minutes to finish half the jar. Since both of them are eating simultaneously, how many minutes would it take them to empty the jar?

Options: A. 4, B. 5, C. 6, D. 7, E. 8

Answer: Let's think step by step.

Model Output

P's rate is 1 jar / 10 minutes = 0.1 [jar/min].
 Q's rate is 1 jar / 5 minutes = 0.1 [jar/min].
 P's rate + Q's rate = 0.1 + 0.2 = 0.3 [jar/min].
 $t = 1 / 0.3 = 3.33$ minutes.
Final Answer: A. 4 ❌

Ours-PFS

Model Input

Re-evaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning.

Initial Thoughts

Question: Two friends are eating a jar full of candies. Had P eaten alone, it would have taken him 10 minutes to finish the candies in the jar. Had Q eaten alone, it would have taken her 5 minutes to finish half the jar. Since both of them are eating simultaneously, how many minutes would it take them to empty the jar?

Options: A. 4, B. 5, C. 6, D. 7, E. 8

Your Previous Answer: ... 3.33 minute **Final Answer: A. 4**

Principles

Principles for Avoiding Mistakes: ...To avoid similar mistakes, always ensure to **verify** calculations and check if the final answer aligns with the provided options.

Principles for Achieving Correct Answers: ...For similar problems, always **determine the rates** of each contributor, **combine them**, and use the reciprocal to find the total time.

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

Retrieved Principles

Principles for Avoiding Mistakes: ...To avoid similar mistakes, always ensure to verify calculations and check if the final answer aligns with the provided options.

Principles for Achieving Correct Answers: ...For similar problems, always determine the rates of each contributor, combine them, and use the reciprocal to find the total time.

Model Output

Step 1: **Determine P's Eating Rate**
 ...

Step 2: **Determine Q's Eating Rate**
 ...

Step 3: **Combine Their Rates**
 When P and Q eat together, their combined rate is:
 Combined rate = 0.1 + 0.1 = 0.2 [jar/min].
 $t = 1 / 0.2 = 5$ minutes.
Final Answer: B. 5 ✅

Figure 1: Prompt examples for each method. Our approach (Ours-PFS) refines Zero-Shot Prompting (Plan and Solve) by injecting retrieved reasoning principles to improve inference.

on vast datasets, require numerous LLM calls, and result in high inference costs and significant computational overhead.

In contrast, when representative Few-Shot examples are available, inference performance can be improved by leveraging those examples directly. LEAP (Zhang et al., 2024) addresses the problem by extracting “reasoning principles”—strategies useful for solving similar problems—by analyzing the differences between correct and incorrect answers based on existing Few-Shot examples, although it assumes that representative Few-Shot examples are already available. In real-world scenarios, carefully curated few-shot exemplars are rarely available, and identifying representative ones from raw data is typically infeasible without significant manual effort.

To overcome these limitations, we propose methods that automatically extract and apply reasoning principles without relying on pre-selected Few-Shot examples. Specifically, we run Zero-Shot Prompting on training data without curated examples, analyze both correct and incorrect reason-

ing paths, and extract principles that capture the essence of correct reasoning or help avoid mistakes. Afterward, for test questions, we retrieve relevant principles and inject them into the model’s inference stage. Our approach focuses on selectively injecting reasoning principles into the parts of the inference process where improvement is most likely—based on similarity to previously observed successes and failures. This design is lightweight yet flexible, improving accuracy on multi-step or complex tasks without incurring the extensive exploration overheads or relying on representative Few-Shot examples, as required by prior methods.

As illustrated in Figure 1, both Zero-Shot and Few-Shot Prompting originally produced incorrect reasoning for a math question. However, by injecting a principle such as “*determine the rates of each contributor; combine them,*” our approach guided the model toward the correct answer.

Contributions. The key contributions of this paper are threefold:

- **Automatic principle extraction.** We intro-

duce a procedure to analyze the reasoning processes on labeled QA data and distill generalizable principles—both from correct answers to reinforce good reasoning and from incorrect ones to avoid common pitfalls.

- **Dynamic principle application.** We propose a retrieval-based mechanism that retrieves relevant principles based on the similarity of reasoning processes, enabling the model to correct or refine its reasoning at inference time.
- **Empirical validation.** Through experiments on benchmarks (GPQA (Rein et al., 2024), MMLU-Pro (Wang et al., 2024), AQuA (Ling et al., 2017), OpenBookQA (Mihaylov et al., 2018)), We demonstrate that our method outperforms the Zero-Shot and Few-Shot baselines by up to 2% in absolute accuracy for complex tasks.

2 Related Work

In this section, we first describe the methods and challenges involved in Zero-Shot and Few-Shot Prompting. Next, we discuss methods aimed at the automatic design of input prompts that include Few-Shot examples. We summarize the challenges of each method and describe their relationship with the proposed approach.

2.1 Zero-Shot and Few-Shot Prompting

Zero-Shot Prompting solves tasks by providing instructional prompts (Kojima et al., 2022). Plan and Solve (Wang et al., 2023), an evolution of Zero-Shot-CoT, generates a plan before reasoning, achieving better accuracy for complex tasks. However, these methods still struggle with tasks requiring task-specific reasoning strategies due to the lack of specific examples.

Few-Shot Prompting provides task-specific examples to guide reasoning (Brown et al., 2020). While effective, designing these examples can be time-consuming and requires domain expertise, as the method is highly sensitive to example format and order (Zhao et al., 2021; Liu et al., 2023). This has led to a growing demand for automated prompt-design methods to reduce reliance on manual effort (Li et al., 2025).

2.2 Automatic Prompt Design Methods Incorporating Few-Shot Examples

Methods to automate prompting with Few-Shot examples include APE (Zhou et al., 2022) and

DSPy (Khattab et al., 2023). These methods aim to optimize the input prompt by leveraging LLMs to iteratively refine its components. Both methods, however, involve exploratory optimization guided by evaluation data, and typically require numerous LLM calls, which can result in high inference costs.

In contrast, LEAP (Zhang et al., 2024) extends standard Few-Shot prompting by having the model intentionally generate incorrect answers, typically using high-temperature sampling. The model then reflects on the difference between its mistaken outputs and the correct answers, extracting generalizable reasoning principles. These principles, expressed in natural language, are added to the prompt and used in subsequent inference, enabling the model to improve its reasoning performance.

2.3 Our Position

In this study, we propose a novel approach that, in environments where representative Few-Shot examples are difficult to identify or unavailable, executes the reasoning process once and, based on the evaluation of that reasoning process with known correct answers, extracts principles that lead to correct answers and avoid incorrect ones. While LEAP (Zhang et al., 2024) also learns from mistakes to derive “reasoning principles,” it requires representative Few-Shot exemplars to seed the error analysis. In contrast, our approach neither presupposes nor depends on carefully selected Few-Shot examples, allowing it to operate in a more general Zero-Shot or limited-resource setting. Furthermore, the extracted principles are retrieved based on the similarity between the reasoning processes of the training and test data, rather than being applied en masse to all test records. This design accommodates a broader range of real-world tasks where curated examples may be difficult to obtain.

3 Proposed Method

In this study, we propose a novel approach consisting of two phases. First, in the **training phase**, QA examples from the training data are processed using Plan and Solve inference with an LLM. The resulting reasoning processes are analyzed to extract common reasoning principles from both successful and failed cases. Second, in the **testing phase**, these extracted principles are applied to test data to improve the reasoning process. Figure 2 illustrates the system model of our approach, and Figure 1

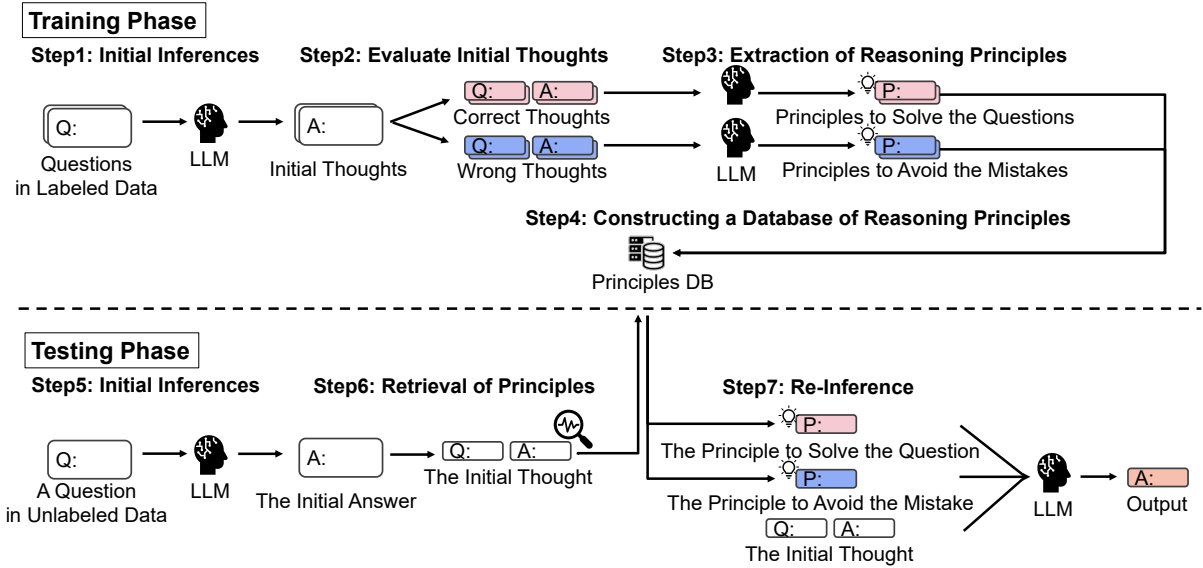


Figure 2: Overview of our method. In the training phase, reasoning principles are extracted from correct and incorrect inferences on labeled data. In the testing phase, relevant principles are retrieved and injected to refine the model’s initial reasoning.

shows sample input and output prompts for the baseline methods (Zero-Shot Prompting, Few-Shot Prompting) and for our method (Ours-PFS).

3.1 Training Phase

In the training phase, the following four steps are executed:

Step 1: Initial Inference. For each question with a known correct answer, Plan and Solve is applied to generate a detailed reasoning process. This reasoning trace is then used to extract useful principles—such as strategies that led to correct answers or helped avoid specific mistakes—and is illustrated as Initial Thoughts in Figure 1.

Step 2: Evaluate Initial Thoughts. The output answer is compared with the known correct answer, and each case is classified as either correct or incorrect.

Step 3: Extraction of Reasoning Principles. In both correct and incorrect cases, the LLM receives the question, the reasoning process, and the correct answer. The LLM is then instructed with prompts such as:

For correct inferences:

“Summarize the verified reasoning that leads to the CORRECT ANSWER in exactly one sentence. Provide exactly one sentence that gives a general strategy for solving similar problems.”

For incorrect inferences:

“Give the correct reasoning that leads to the CORRECT ANSWER in exactly one sentence. Provide exactly one sentence that gives a general strategy for avoiding similar mistakes.”

These instructions enable the automatic extraction of reasoning principles that can be generalized to similar problems.

Step 4: Constructing a Database of Reasoning Principles. Each extracted principle is stored in a database along with the corresponding question and its reasoning trace. To enable similarity-based retrieval, we concatenate the question and its CoT reasoning into a single text string and compute its vector representation using an embedding model. These vectors are then stored in a vector database as keys to retrieve relevant principles at inference time.

3.2 Testing Phase

The testing phase consists of three steps:

Step 5: Initial Inference. Each test sample is processed with Plan and Solve, yielding an initial reasoning process.

Step 6: Retrieval of Principles. The test sample and its initially generated CoT are concatenated into a single text and then embedded using the same embedding model as in the training phase.

We perform a similarity search via cosine similarity against the pre-constructed database of reasoning principles, retrieving the top- k most similar entries. Each entry in the database corresponds to a principle—extracted either from correct or incorrect examples—and retrieved based on the concatenation of the question and its CoT reasoning. By injecting these retrieved principles, the model can reinforce effective strategies from correct examples and avoid common pitfalls from incorrect examples before performing the final inference step.

Step 7: Re-Inference. The retrieved reasoning principles are appended to the prompt alongside the initial CoT output. Specifically, we place the principles in a short natural-language paragraph after the model’s first reasoning trace and instruct the model to “Re-evaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning.” We then ask the LLM to perform inference again. This step leverages the extracted principles both to avoid pitfalls identified in previous failures and to reinforce successful reasoning strategies.

4 Experiments and Results

To validate the effectiveness of our method, we conducted experiments in which the reasoning principles were extracted from each LLM’s inference process on **training** data, and then applied to correct the reasoning on **test** data. In these experiments, we primarily focus on two LLMs, GPT-4o and GPT-4o-mini (Hurst et al., 2024), which are representative of the available models and widely used in practice.

To measure text similarity, each text is converted into an embedding representation via the Sentence Transformer (all-MiniLM-L6-v2) (Reimers and Gurevych, 2020), and the cosine similarity is calculated. We evaluate the baseline methods (Zero-Shot, Few-Shot, and Plan-Solve) and our proposed approach (Ours-PF, Ours-PS, Ours-PFS) under these models.

4.1 Benchmarks

We selected these four benchmarks to capture a range of reasoning challenges: GPQA and MMLU-Pro emphasize multi-step inference across scientific and academic domains—tasks that are particularly challenging for Zero-Shot prompting. AQuA focuses on mathematical reasoning, while OpenBookQA targets common-sense knowledge:

Table 1: Overview of the benchmarks used in our experiments, along with the number of samples used in the training and testing phases.

Data set	Task Type	Train	Test
GPQA	Advanced	250	198
MMLU-Pro	Advanced	560	1,120
AQuA	Math	500	254
OpenBookQA	General	500	500

GPQA (GPQA-diamond and GPQA-main):

GPQA is a challenging benchmark in biology, physics, and chemistry (Rein et al., 2024). Each question is presented in a multiple-choice format with four options, only one of which is correct. For training, we used GPQA-main, which includes easier questions. For testing, we used GPQA-diamond, a more difficult subset validated by experts and characterized by low accuracy from non-experts. To ensure a clear distinction between training and test sets, any overlap with GPQA-diamond was removed from the training data.

MMLU-Pro: A challenging benchmark composed of questions derived from academic exams and textbooks, spanning a wide range of subjects grouped into 14 categories (Wang et al., 2024). Each question has 10 answer choices, only one of which is correct. Training and test samples are randomly selected in equal numbers from each category.

AQuA: A benchmark featuring algebraic word problems accompanied by rationales (Ling et al., 2017). Each question includes five multiple-choice options, with only one correct answer.

OpenBookQA: A benchmark consisting of questions that require broad common knowledge (Mihaylov et al., 2018). Each question provides four answer choices, with a single correct answer.

An overview of each benchmark and the number of samples used in the training and testing phases is provided in Table 1.

4.2 Comparison Methods

To ensure the reliability of the results, each experiment on the training data was repeated three times with different random seeds, and the average accuracy was computed. Accuracy is used as the evaluation metric, since all benchmarks consist of single-answer multiple-choice questions with class balance. In addition, metrics like F1 score are less suitable, as the model may sometimes produce non-choice or abstention responses. In all experi-

Table 2: Average accuracy (%) of different inference methods across four benchmarks for GPT-4o and GPT-4o-mini (3-shot). The best-performing method for each benchmark is highlighted in bold.

Benchmark	Task Type	Model	Zero-Shot	Plan-Solve	Few-Shot	Ours-PF	Ours-PS	Ours-PFS
GPQA	Advanced	GPT-4o	49.8	49.0	43.9	51.0	51.2	50.2
		GPT-4o-mini	41.8	44.8	39.6	45.8	47.1	45.6
MMLU-Pro	Advanced	GPT-4o	74.9	75.2	61.3	75.5	75.6	75.4
		GPT-4o-mini	64.4	64.6	63.1	65.1	65.0	65.1
AQuA	Math	GPT-4o	82.0	82.7	44.8	83.5	84.0	84.5
		GPT-4o-mini	78.6	80.2	78.1	80.4	81.5	81.4
OpenBookQA	General	GPT-4o	96.1	96.5	96.1	96.7	96.1	96.7
		GPT-4o-mini	94.3	94.5	92.6	94.3	94.3	94.6

ments, the *temperature* of the LLM was set to 0 to minimize variability in inference outcomes.

In these experiments, we set $k = 3$ for the number of retrieved principles in each case. Specifically, for principle retrieval, we use three principles for each of Ours-PF and Ours-PS, and three principles each (three successes and three failures) for Ours-PFS. For Few-Shot, we retrieve three example QAs based on question similarity, selecting the top three examples from the training data with the highest similarity to the test question, which were answered correctly by Zero-Shot inference. Additionally, the impact of k on inference performance will be discussed in Section 5.

In this section, we compare our method against several baselines to evaluate its performance. Among the methods introduced in Section 2, we exclude APE, DSPy, and LEAP from comparison, as our setting does not assume access to representative Few-Shot exemplars or allow for iterative prompt optimization.

- **Zero-Shot Prompting (Zero-Shot):** The baseline method where only the instruction “Let’s think step by step.” is appended at the end of the task prompt.
- **Plan and Solve (Plan-Solve):** An enhanced baseline method, where the instruction to devise a plan and solve the problem step by step is added to promote structured problem-solving. This method serves as the initial thought process in our proposed approach, helping to solve problems in a clear and systematic way.
- **Few-Shot Prompting (Few-Shot):** For each test question, we retrieve the most similar training example—based on cosine similarity of Sentence Transformer embeddings of the question text—among those for which

Zero-Shot inference yielded the correct answer. The question and gold answer from this retrieved question is then provided in the prompt to guide inference. This method serves to demonstrate the effect of using exemplars as opposed to extracted principles.

- **Proposed Method:** As depicted in Figure 1, our approach incorporates the extracted reasoning principles to prompt local re-generation of the reasoning process. This method is evaluated in three variants:
 - **Ours-PF (Principles from Failures):** Uses principles derived from past failed cases to trigger correction in the reasoning process.
 - **Ours-PS (Principles from Successes):** Applies principles extracted from past successful inferences to reinforce correct reasoning.
 - **Ours-PFS (Principles from Failures and Successes):** Combines the principles from both failures and successes, simultaneously correcting and reinforcing the reasoning process.

For reproducibility, we include full prompt templates and sample outputs in the Appendix.

4.3 Evaluation Results

Table 2 summarizes the average accuracy (in percentage) for each benchmark and model across the different methods.

Compared with the strongest baseline, Plan-Solve, the gains are largest on GPQA, where Ours-PS raises GPT-4o from 49.0 % to 51.2 % and GPT-4o-mini from 44.8 % to 47.1 % (both > 2 pp). On MMLU-Pro the margin is modest but consistent (up to +0.5 pp), while on AQuA it ranges from +0.2 pp to +1.8 pp, with the larger improvement

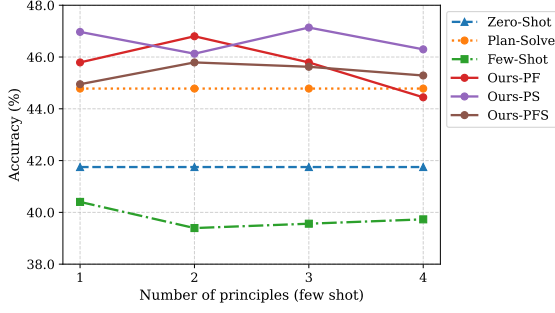


Figure 3: Accuracy of GPQA and GPT-4o-mini as a function of few-shot count k .

observed for GPT-4o. For OpenBookQA, which relies more on factual recall than multi-step reasoning, all methods cluster within 0.2 pp of the baseline.

The optimal variant is task-dependent. Ours-PS, which reinforces reasoning patterns found in correct traces, is best on the two advanced benchmarks: it tops GPQA for both models and achieves the highest score on MMLU-Pro with GPT-4o. Ours-PFS, which blends principles from successes and failures, excels on the Math benchmark AQuA. Ours-PF, based solely on failure analysis, occasionally matches or narrowly beats the other variants. In contrast, the common-knowledge benchmark OpenBookQA exhibits virtually no benefit from principle injection. These results show that injecting reasoning principles is beneficial, but the mix of success- and failure-derived guidance that works best varies with the domain and complexity of the task.

Additionally, a closer look at the Few-Shot baseline reveals that, with the simple cosine-similarity retrieval used here, Few-Shot is actually worse than Zero-Shot on every benchmark. It is underscoring that uncured exemplars alone are insufficient for most tasks.

5 Discussion

5.1 Analysis on Multiple Benchmarks

Our experimental results show that principle injection produces the largest gain on the challenging GPQA set (about +2 pp), a consistent but smaller gain on MMLU-Pro (around +0.5 pp for both models), and a modest yet measurable gain on AQuA, ranging from +0.2 to +1.0 pp depending on the model. These patterns indicate that injecting reasoning principles is especially beneficial for complex multi-step tasks, while still offering some im-

provement even when the baseline accuracy is already high. For example, in the MMLU-Pro benchmark, our method corrected a reasoning error in a likelihood ratio test question. Initially, the model incorrectly assessed the relationship between the models' log-likelihoods. After applying the principle of verifying model parameters, and encouraging a deeper understanding of the likelihood ratio test, the model recalculated correctly and identified the accurate answer.

Meanwhile, no improvement is observed on OpenBookQA, which relies heavily on common knowledge rather than logical inference. Our method performed similarly to Zero-Shot, suggesting that tasks requiring common-sense knowledge, which the model already struggles with in Zero-Shot, cannot be resolved by adjusting the reasoning process alone. Detailed examples of both the MMLU-Pro and OpenBookQA cases discussed above are provided in the Appendix.

Although previous studies often report that Few-Shot prompts can surpass Zero-Shot inference, our experiments tell a different story. For each test query we retrieved the training question closest in embedding space that Zero-Shot had already answered correctly; these uncured exemplars may resemble the target problem superficially, but they seldom capture the underlying reasoning structure required to solve it. This finding exposes a practical limitation: the accuracy gains of Few-Shot prompting hinge on expert curation—an asset that is costly to reproduce in real deployments. By contrast, our principle-injection method still yields consistent improvements even when only raw, uncured QA pairs are available.

5.2 Effect of varying k on performance

To examine how the number of retrieved principles influences accuracy, we varied the retrieval depth k . For GPQA, the results from GPT-4o-mini with $k = 1$ to $k = 4$ are shown in Figure 3. We observe that as k increases, the performance becomes more stable, but the accuracy does not significantly increase. This suggests that while smaller values of k might result in ineffective principle retrieval due to a limited number of principles being retrieved, larger values of k can lead to confusion, as the model may struggle to determine which principles to follow when too many are presented. Therefore, although larger values of k bring more principles, it does not necessarily improve the accuracy significantly.

In this study, we opted for $k = 3$ as a balance between the number of principles retrieved and the stability of performance. However, the optimal value of k may vary depending on how well the database of principles is constructed and how effective the retrieval mechanism is at selecting relevant principles. The results for other benchmarks concerning GPT-4o-mini with varying k (from $k = 1$ to $k = 4$) are provided in the Appendix.

5.3 Category-Level Analysis on MMLU-Pro

As shown in Table 2, our method yields accuracy improvements across some benchmarks. To better understand its robustness and generalizability, we conduct a fine-grained analysis on the MMLU-Pro benchmark, which covers a wide range of academic subjects (e.g., chemistry, business, history) and presents diverse reasoning challenges. Given its broad coverage, MMLU-Pro is particularly well-suited for examining how our method performs across different categories.

Figure 4 compares the accuracy of Ours-PFS with Plan-Solve for each category, using GPT-4o-mini and the full range of $k = 1$ to $k = 4$ across all seeds. In this figure, categories where the horizontal axis is positive indicate areas where our proposed method provides an improvement over Plan-Solve. We observe that Ours-PFS consistently boosts performance in logic-heavy domains such as business, physics, and engineering. In contrast, categories like philosophy show slightly negative gains, suggesting that these areas may rely more on factual recall or specialized knowledge rather than step-by-step logical reasoning. This implies that certain question types might require additional knowledge or alternative prompting strategies.

6 Conclusion

In this study, we proposed a novel in-context learning approach that applies reasoning principles extracted from LLM inferences on training data without relying on task-specific Few-Shot examples to improve reasoning on test data. Experimental results demonstrated that our method achieved an accuracy improvement of up to 2% on complex reasoning tasks such as GPQA compared to Zero-Shot, Plan and Solve, and Few-Shot methods. Moreover, the effectiveness of our approach on both GPT-4o and GPT-4o-mini suggests its potential applicability to a wider range of LLMs. In future work, we plan to further investigate the generality of our

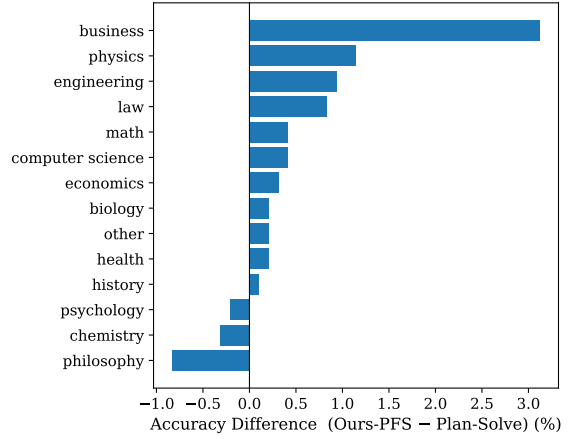


Figure 4: Accuracy difference between *Ours-PFS* and *Plan-Solve* across MMLU domains ($k = 1 \sim 4$ shot).

method by applying it to diverse tasks and models, including the reasoning trajectories and strategies of AI agents involved in complex decision-making processes, and to refine our approach for even more robust correction of reasoning processes. An important future direction is to explore whether the required amount of training data can be further reduced for tasks with a limited variety of reasoning principles. In addition, verifying the potential for domain transfer remains an open challenge, and we aim to evaluate how well the extracted principles generalize across different domains.

7 Limitations

One practical limitation of our method is the increased inference cost compared to a standard Zero-Shot CoT approach. During both the training and testing phases, two inference steps are required: (1) an initial inference to generate the reasoning trace, and (2) a second step to extract principles or refine the reasoning with retrieved principles.

However, the principle extraction step in training only needs to be performed once per domain or benchmark. In real-world scenarios where the domain remains consistent, these principles can be reused, reducing the additional cost to just one inference per question during testing. While the initial training phase incurs some overhead, the method is more scalable than alternatives like Few-Shot prompting, which require curated examples or optimization.

In addition, it assumes access to a modest amount of task data to extract useful reasoning patterns. In our experiments, around 500 exam-

ples per benchmark were sufficient to yield gains. However, in extremely low-resource settings where such examples are unavailable, performance may be more limited.

Another limitation of our method is that it assumes access to labeled QA data for principle extraction. In many real-world scenarios, such QA pairs are readily available. For example, they can be obtained from question–answer logs in customer service systems, community Q&A platforms, or existing QA benchmarks. This makes the assumption practically reasonable.

Ethics Statement

This work uses only publicly available benchmarks and LLMs, and does not involve any human subjects or private data.

Supplementary Materials Availability Statement

All benchmarks used in our experiments are publicly available open-source benchmarks. This experiment was conducted through prompt-driven experiments, and the full text of the prompts necessary for reproduction is provided in the Appendix.

References

- 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, Online. Curran Associates, Inc.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. [Gpt-4o system card](#). *arXiv preprint arXiv:2410.21276*.
- Omar Khattab et al. 2023. [Dspy: Compiling declarative language model calls into self-improving pipelines](#). *arXiv preprint arXiv:2310.03714*.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213, New Orleans, Louisiana, USA. Curran Associates, Inc.
- Yanis Labrak, Mickael Rouvier, and Richard Dufour. 2024. [A zero-shot and few-shot study of instruction-finetuned large language models applied to clinical and biomedical tasks](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2049–2066, Torino, Italia. ELRA and ICCL.
- Wenwu Li, Xiangfeng Wang, Wenhao Li, and Bo Jin. 2025. [A survey of automatic prompt engineering: An optimization perspective](#). *arXiv preprint arXiv:2502.11560*.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program induction by rationale generation: Learning to solve and explain algebraic word problems](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *ACM Comput. Surv.*, 55(9).
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Saeel Sandeep Nachane, Ojas Gramopadhye, Prateek Chanda, Ganesh Ramakrishnan, Kshitij Sharad Jadhav, Yatin Nandwani, Dinesh Raghu, and Sachindra Joshi. 2024. [Few shot chain-of-thought driven reasoning to prompt LLMs for open-ended medical question answering](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 542–573, Miami, Florida, USA. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4512–4525, Online. Association for Computational Linguistics.

- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. [On second thought, let’s not think step by step! bias and toxicity in zero-shot reasoning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4454–4470, Toronto, Canada. Association for Computational Linguistics.
- Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. 2022. [On the effect of pretraining corpora on in-context learning by a large-scale language model](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5168–5186, Seattle, United States. Association for Computational Linguistics.
- Chaozheng Wang, Yuanhang Yang, Cuiyun Gao, Yun Peng, Hongyu Zhang, and Michael R Lyu. 2022. [No more fine-tuning? an experimental evaluation of prompt tuning in code intelligence](#). In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 382–394, Singapore. Association for Computing Machinery.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). *arXiv preprint arXiv:2406.01574*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Noam Wies, Yoav Levine, and Amnon Shashua. 2023. [The learnability of in-context learning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 36637–36651, New Orleans, Louisiana, USA. Curran Associates, Inc.
- Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024. [In-context principle learning from mistakes](#). *arXiv preprint arXiv:2402.0540*.
- Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. [Calibrate before use: Improving few-shot performance of language models](#). *arXiv preprint arXiv:2102.09690*.
- Yongchao Zhou et al. 2022. [Large language models are human-level prompt engineers](#). *arXiv preprint arXiv:2211.01910*.

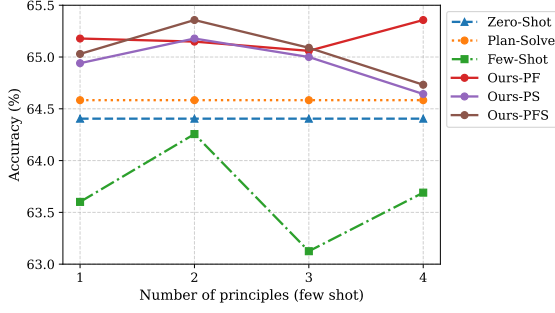


Figure 5: Accuracy of MMLU-Pro for GPT-4o-mini as a function of the number of retrieved principles k .

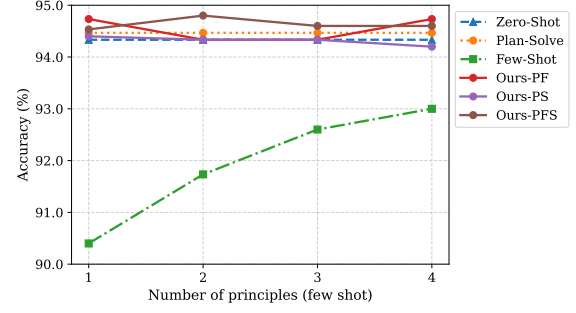


Figure 7: Accuracy of OpenBookQA for GPT-4o-mini as a function of the number of retrieved principles k .

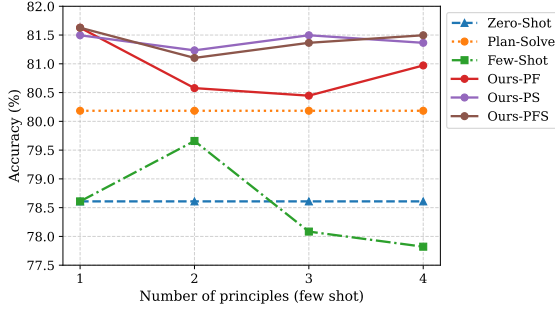


Figure 6: Accuracy of AQuA for GPT-4o-mini as a function of the number of retrieved principles k .

task. While Few-Shot has relatively low accuracy to begin with, increasing k clearly improves performance. This suggests that for tasks like OpenBookQA, which rely on general knowledge, the Few-Shot method might enhance answer accuracy by incorporating relevant knowledge from the QAs. However, it remains lower in accuracy compared to the other methods.

A Additional Results for Varying k

Here, we present the accuracy experiments for GPT-4o-mini with varying k values from 1 to 4, which were not included in the main text, for each benchmark.

A.1 MMLU-Pro

As shown in Figure 5, increasing k stabilizes performance on the MMLU-Pro benchmark. However, no significant changes are observed with varying k , and the performance of Ours methods slightly outperforms Plan-Solve.

A.2 AQuA

Figure 6 shows that as k increases, the performance on AQuA remains relatively stable across methods, with only slight variations. The accuracy does not fluctuate significantly with varying k , and Ours-PS and Ours-PFS maintains a consistent performance, slightly outperforming the other methods.

A.3 OpenBookQA

Figure 7 demonstrates the stable performance on OpenBookQA across varying k . Most methods show no significant change in accuracy, reflecting the limited potential for improvement in this

B Prompt Templates

Here, we provide template prompts used for the extraction and application of reasoning principles. In each template, placeholders enclosed in curly braces (e.g., {question}) indicate elements that will be dynamically inserted based on the input data.

B.1 Prompt to Extract Reasoning Principles

For principle extraction, the prompt requires the training question, the correct answer, and the initial reasoning trace as inputs. We design two separate prompts depending on whether the initial reasoning was correct or incorrect:

- For **correct inferences**, the prompt asks the model to summarize the reasoning path that led to the correct answer and to extract a general strategy for solving similar problems.
- For **incorrect inferences**, the prompt instructs the model to reconstruct the correct reasoning and derive a strategy to avoid similar mistakes in the future.

This distinction allows the system to capture both positive and negative reasoning patterns that can be generalized across examples, as illustrated in the following prompts.

Prompt to Extract Reasoning Principles for Correct Thoughts

You are a specialist tutor.

NOTE: The ATTEMPTED ANSWER may be partially or entirely incorrect. Reconstruct the correct reasoning from the QUESTION and CORRECT ANSWER; reuse parts of the attempt only after independent verification.

QUESTION:
{question}

ATTEMPTED ANSWER (may contain errors):
{initial thoughts}

CORRECT ANSWER:
{correct answer}

—

Write a compact solution for future learners:

- Summarize the verified reasoning that leads to the CORRECT ANSWER in exactly one sentence.
- Provide exactly one sentence that gives a general strategy for solving similar problems.

Prompt to Extract Reasoning Principles for Wrong Thoughts

You are a specialist tutor.

NOTE: The ATTEMPTED ANSWER may be partially or entirely incorrect. Reconstruct the correct reasoning from the QUESTION and CORRECT ANSWER; reuse parts of the attempt only after independent verification.

QUESTION:
{question}

ATTEMPTED ANSWER (may contain errors):
{initial thoughts}

CORRECT ANSWER:
{correct answer}

—

Write a compact solution for future learners:

- Give the correct reasoning that leads to the CORRECT ANSWER in exactly one sentence.
- Provide exactly one sentence that gives a general strategy for avoiding similar mistakes.

B.2 Prompt to Apply Reasoning Principles at Test Time

The following prompt is used in the inference phase of **Ours-PF**, **Ours-PS**, and **Ours-PFS**. It presents the original question, the model’s previous answer, and the retrieved reasoning principles. The model is then instructed to revise its reasoning based on these principles. The core prompt structure is shared across the three variants. **Ours-PF** uses only principles derived from failed examples, **Ours-PS** uses only those derived from successful examples, and **Ours-PFS** uses both.

Prompt for Re-Inference with Injected Principles

Re-evaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning.

Select the single best option (A, B, C, ...). 'None' is not allowed.

Question:
{question}

Your Previous Answer:
{initial thoughts}

Principles for Avoiding Mistakes:

<F-1> {failure_principle_1}

<F-2> {failure_principle_2}

...

Principles for Achieving Correct Answers:

<S-1> {success_principle_1}

<S-2> {success_principle_2}

...

Answer: Let’s first understand the problem and devise a plan to solve the problem.
Then, let’s carry out the plan and solve the problem step by step.

C Prompt Examples on MMLU-Pro benchmark

This appendix provides the exact prompt templates used in our experiments. For the MMLU-Pro benchmark, we include complete examples (both model input and output) for Plan-Solve and Ours-PFS. In addition, we provide the model input prompts for the other methods (Zero-Shot, Few-Shot, Ours-PF, and Ours-PS) as supplementary material, assuming the case where the number of retrieved principles or Few-Shot examples is $k = 1$.

C.1 Plan-Solve

We present both the input prompt and the corresponding model output below as a representative example.

MMLU-Pro: Plan-Solve – Model Input Prompt Example

Select the single best option (A, B, C, D). 'None' is not allowed.

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Options: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Answer: Let’s first understand the problem and devise a plan to solve the problem.
Then, let’s carry out the plan and solve the problem step by step.

MMLU-Pro: Plan-Solve – Model Output Prompt Example

To solve the problem, we need to analyze each of the statements provided in the context of the likelihood ratio test (LRT) for a GARCH(2,2) model being tested against a restricted model with constant conditional variance.

Step 1: Understanding the Likelihood Ratio Test The likelihood ratio test compares the goodness of fit of two models: the unrestricted model (GARCH(2,2)) and the restricted model (constant variance). The test statistic is calculated as:

$$LR = -2 \times (\text{Log-Likelihood of Restricted Model} - \text{Log-Likelihood of Unrestricted Model})$$

Under the null hypothesis, this test statistic follows a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters estimated in the two models.

Step 2: Analyzing Each Statement

Statement A: "The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis."

- The GARCH(2,2) model has more parameters than the constant variance model. The constant variance model has 1 parameter (the mean), while the GARCH(2,2) model has 4 parameters (mean, two GARCH parameters, and one variance parameter). Therefore, the difference in parameters is 3. Thus, the test statistic will follow a chi-squared distribution with 3 degrees of freedom, not 2. **This statement is false.**

Statement B: "The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model."

- The log-likelihood of the unrestricted model (GARCH(2,2)) should be greater than or equal to that of the restricted model (constant variance) because the unrestricted model has more parameters and can fit the data better. **This statement is true.**

Statement C: "If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2."

- The test statistic is calculated as:

$$LR = -2 \times (-118.4 - (-112.3)) = -2 \times (-118.4 + 112.3) = -2 \times (-6.1) = 12.2$$

This calculation is correct. **This statement is true.**

Statement D: "The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value."

- The likelihood ratio test does not compare slopes; it compares the log-likelihood values of the two models. The test statistic is based on the difference in log-likelihoods, not on the slopes. **This statement is false.**

Step 3: Conclusion Based on the analysis, the true statements are B and C. However, since we can only select one option, we need to choose the best one.

- **B** is a general statement about the log-likelihood values, which is always true. - **C** is a specific calculation that is also true.

Since both B and C are true, but B is a broader statement about the nature of the models, we will select **B** as the best option.

Final Answer: **B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model.** ✗ (Correct: C)

C.2 Ours-PFS (Principles from Failures and Successes)

As with Plan-Solve, we provide both the model input and output to illustrate how our method (Ours-PFS) operates in practice.

MMLU-Pro: Ours-PFS – Model Input Prompt Example

Reevaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning. Select the single best option (A, B, C, D, E, F, G, H, I, J). 'None' is not allowed.

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Options: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Your Previous Answer: To solve the problem, we need to analyze each of the statements provided in the context of the likelihood ratio test (LRT) for a GARCH(2,2) model being tested against a restricted model with constant conditional variance.

Step 1: Understanding the Likelihood Ratio Test The likelihood ratio test compares the goodness of fit of two models: the unrestricted model (GARCH(2,2)) and the restricted model (constant variance). The test statistic is calculated as:

$$LR = -2 \times (\text{Log-Likelihood of Restricted Model} - \text{Log-Likelihood of Unrestricted Model})$$

Under the null hypothesis, this test statistic follows a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters estimated in the two models.

Step 2: Analyzing Each Statement

Statement A: "The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis."

- The GARCH(2,2) model has more parameters than the constant variance model. The constant variance model has 1 parameter (the mean), while the GARCH(2,2) model has 4 parameters (mean, two GARCH parameters, and one variance parameter). Therefore, the difference in parameters is 3. Thus, the test statistic will follow a chi-squared distribution with 3 degrees of freedom, not 2. **This statement is false.**

Statement B: "The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model."

- The log-likelihood of the unrestricted model (GARCH(2,2)) should be greater than or equal to that of the restricted model (constant variance) because the unrestricted model has more parameters and can fit the data better. **This statement is true.**

Statement C: "If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2."

- The test statistic is calculated as:

$$LR = -2 \times (-118.4 - (-112.3)) = -2 \times (-118.4 + 112.3) = -2 \times (-6.1) = 12.2$$

This calculation is correct. **This statement is true.**

Statement D: "The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value."

- The likelihood ratio test does not compare slopes; it compares the log-likelihood values of the two models. The test statistic is based on the difference in log-likelihoods, not on the slopes. **This statement is false.**

Step 3: Conclusion Based on the analysis, the true statements are B and C. However, since we can only select one option, we need to choose the best one.

- **B** is a general statement about the log-likelihood values, which is always true. - **C** is a specific calculation that is also true.

Since both B and C are true, but B is a broader statement about the nature of the models, we will select **B** as the best option.

Final Answer: **B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model. ✗ (Correct: C)

Principles for Avoiding Mistakes:

⟨F-1⟩ To test whether the VAR(6) can be restricted to a VAR(4), use the likelihood ratio test statistic calculated as $LR = -2 \cdot \log\left(\frac{0.0049}{0.0036}\right)$, which yields a value of approximately 77.07.

To avoid similar mistakes, ensure to correctly apply the likelihood ratio test formula and verify each step of the calculation for accuracy.

Principles for Achieving Correct Answers:

⟨S-1⟩ The statement that is NOT true is D, as the critical value for the rejection region actually decreases when the degrees of freedom decrease, contrary to what is claimed.

A general strategy for solving similar problems is to analyze each statement in the context of the properties and characteristics of the relevant statistical distribution.

Answer: Let's first understand the problem and devise a plan to solve the problem.

Then, let's carry out the plan and solve the problem step by step.

MMLU-Pro: Ours-PFS – Model Output Prompt Example

Let's reevaluate the question and the statements provided regarding the likelihood ratio test (LRT) for a GARCH(2,2) model against a restricted model with constant conditional variance.

Step 1: Understanding the Likelihood Ratio Test The likelihood ratio test compares the goodness of fit of two models: the unrestricted model (GARCH(2,2)) and the restricted model (constant variance). The test statistic is calculated as:

$$LR = -2 \times (\text{Log-Likelihood of Restricted Model} - \text{Log-Likelihood of Unrestricted Model})$$

Under the null hypothesis, this test statistic follows a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters estimated in the two models.

Step 2: Analyzing Each Statement

Statement A: "The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis."

- The GARCH(2,2) model has 4 parameters (mean, two GARCH parameters, and one variance parameter), while the constant variance model has 1 parameter. The difference in the number of parameters is 3. Therefore, the test statistic will follow a chi-squared distribution with 3 degrees of freedom, not 2. **This statement is false.**

Statement B: "The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model."

- The log-likelihood of the unrestricted model (GARCH(2,2)) should be greater than or equal to that of the restricted model (constant variance) because the unrestricted model has more parameters and can fit the data better. **This statement is false.** The log-likelihood of the unrestricted model is typically higher, not the restricted model.

Statement C: "If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2."

- The test statistic is calculated as:

$$LR = -2 \times (-118.4 - (-112.3)) = -2 \times (-118.4 + 112.3) = -2 \times (-6.1) = 12.2$$

This calculation is correct. **This statement is true.**

Statement D: "The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value."

- The likelihood ratio test does not compare slopes; it compares the log-likelihood values of the two models. The test statistic is based on the difference in log-likelihoods, not on the slopes. **This statement is false.**

Step 3: Conclusion Based on the analysis, the true statement is C.

- **A** is false because the degrees of freedom are 3, not 2. - **B** is false because the unrestricted model typically has a higher log-likelihood. - **C** is true as the calculation of the test statistic is correct. - **D** is false because the test does not compare slopes.

Since we can only select one option and C is the only true statement, the best option is:

Final Answer: **C**. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2. ✓

C.3 Other Methods

Below, we also provide input prompt examples for the other methods used in our experiments. Few-Shot uses the most similar successful example (based on question embedding similarity) as a single in-context demonstration. Ours-PF injects only principles derived from failed cases to correct prior mistakes, while Ours-PS uses only principles from successful cases to reinforce effective reasoning. Ours-PFS, which combines both, is presented earlier with complete input-output examples.

MMLU-Pro: Zero-Shot – Model Input Prompt Example

Select the single best option (A, B, C, D). 'None' is not allowed.

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Options: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Answer: Let's think step by step.

MMLU-Pro: Few-Shot – Model Input Prompt Example

Select the single best option (A, B, C, D). 'None' is not allowed.

Question: Which of the following propositions would be rejected by those who describe themselves as critical legal theorists?

Options: A. Law is unstable., B. Law is determinate., C. Law reflects economic power., D. Law is politics., E. Law is a social construct.

Answer: B

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Answer: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Answer: Let's think step by step.

MMLU-Pro: Ours-PF – Model Input Prompt Example

Reevaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning. Select the single best option (A, B, C, D, E, F, G, H, I, J). 'None' is not allowed.

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Options: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Your Previous Answer: To solve the problem, we need to analyze each of the statements provided in the context of the likelihood ratio test (LRT) for a GARCH(2,2) model being tested against a restricted model with constant conditional variance.

Step 1: Understanding the Likelihood Ratio Test The likelihood ratio test compares the goodness of fit of two models: the unrestricted model (GARCH(2,2)) and the restricted model (constant variance). The test statistic is calculated as:

$$LR = -2 \times (\text{Log-Likelihood of Restricted Model} - \text{Log-Likelihood of Unrestricted Model})$$

Under the null hypothesis, this test statistic follows a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters estimated in the two models.

Step 2: Analyzing Each Statement

****Statement A:**** "The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis."

- The GARCH(2,2) model has more parameters than the constant variance model. The constant variance model has 1 parameter (the mean), while the GARCH(2,2) model has 4 parameters (mean, two GARCH parameters, and one variance parameter). Therefore, the difference in parameters is 3. Thus, the test statistic will follow a chi-squared distribution with 3 degrees of freedom, not 2. ****This statement is false.****

****Statement B:**** "The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model."

- The log-likelihood of the unrestricted model (GARCH(2,2)) should be greater than or equal to that of the restricted model (constant variance) because the unrestricted model has more parameters and can fit the data better. ****This statement is true.****

****Statement C:**** "If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2."

- The test statistic is calculated as:

$$LR = -2 \times (-118.4 - (-112.3)) = -2 \times (-118.4 + 112.3) = -2 \times (-6.1) = 12.2$$

This calculation is correct. ****This statement is true.****

****Statement D:**** "The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value."

- The likelihood ratio test does not compare slopes; it compares the log-likelihood values of the two models. The test statistic is based on the difference in log-likelihoods, not on the slopes. ****This statement is false.****

Step 3: Conclusion Based on the analysis, the true statements are B and C. However, since we can only select one option, we need to choose the best one.

- ****B**** is a general statement about the log-likelihood values, which is always true. - ****C**** is a specific calculation that is also true.

Since both B and C are true, but B is a broader statement about the nature of the models, we will select ****B**** as the best option.

Final Answer: ****B.** The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model. ****X (Correct: C)**

Principles for Avoiding Mistakes:

(F-1) To test whether the VAR(6) can be restricted to a VAR(4), use the likelihood ratio test statistic calculated as $LR = -2 \cdot \log \left(\frac{0.0049}{0.0036} \right)$, which yields a value of approximately 77.07.

To avoid similar mistakes, ensure to correctly apply the likelihood ratio test formula and verify each step of the calculation for accuracy.

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

MMLU-Pro: Ours-PS – Model Input Prompt Example

Reevaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning. Select the single best option (A, B, C, D, E, F, G, H, I, J). 'None' is not allowed.

Question: Suppose that we are interested in testing the null hypothesis that a GARCH(2,2) model can be restricted to a process with a constant conditional variance using the likelihood ratio test approach. Which of the following statements are true?

Options: A. The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis, B. The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model, C. If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2, D. The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value.

Your Previous Answer: To solve the problem, we need to analyze each of the statements provided in the context of the likelihood ratio test (LRT) for a GARCH(2,2) model being tested against a restricted model with constant conditional variance.

Step 1: Understanding the Likelihood Ratio Test The likelihood ratio test compares the goodness of fit of two models: the unrestricted model (GARCH(2,2)) and the restricted model (constant variance). The test statistic is calculated as:

$$LR = -2 \times (\text{Log-Likelihood of Restricted Model} - \text{Log-Likelihood of Unrestricted Model})$$

Under the null hypothesis, this test statistic follows a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters estimated in the two models.

Step 2: Analyzing Each Statement

****Statement A:**** "The test statistic will follow a chi-squared distribution with 2 degrees of freedom under the null hypothesis."

- The GARCH(2,2) model has more parameters than the constant variance model. The constant variance model has 1 parameter (the mean), while the GARCH(2,2) model has 4 parameters (mean, two GARCH parameters, and one variance parameter). Therefore, the difference in parameters is 3. Thus, the test statistic will follow a chi-squared distribution with 3 degrees of freedom, not 2. ****This statement is false.****

****Statement B:**** "The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model."

- The log-likelihood of the unrestricted model (GARCH(2,2)) should be greater than or equal to that of the restricted model (constant variance) because the unrestricted model has more parameters and can fit the data better. ****This statement is true.****

****Statement C:**** "If the relevant values of the log-likelihood functions are -112.3 and -118.4, the value of the test statistic is 12.2."

- The test statistic is calculated as:

$$LR = -2 \times (-118.4 - (-112.3)) = -2 \times (-118.4 + 112.3) = -2 \times (-6.1) = 12.2$$

This calculation is correct. ****This statement is true.****

****Statement D:**** "The likelihood ratio test compares the slopes of the log-likelihood function at the maximum and at the restricted parameter value."

- The likelihood ratio test does not compare slopes; it compares the log-likelihood values of the two models. The test statistic is based on the difference in log-likelihoods, not on the slopes. ****This statement is false.****

Step 3: Conclusion Based on the analysis, the true statements are B and C. However, since we can only select one option, we need to choose the best one.

- ****B**** is a general statement about the log-likelihood values, which is always true. - ****C**** is a specific calculation that is also true.

Since both B and C are true, but B is a broader statement about the nature of the models, we will select ****B**** as the best option.

Final Answer: ****B.** The value of the log-likelihood function will almost always be bigger for the restricted model than for the unrestricted model. ****X (Correct: C)**

Principles for Achieving Correct Answers:

(S-1) The statement that is NOT true is D, as the critical value for the rejection region actually decreases when the degrees of freedom decrease, contrary to what is claimed.

A general strategy for solving similar problems is to analyze each statement in the context of the properties and characteristics of the relevant statistical distribution.

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

D Prompt Examples on OpenBookQA benchmark

This appendix provides the exact prompt templates used in our experiments on the OpenBookQA benchmark. For OpenBookQA, we include complete examples (both model input and output) for Zero-Shot and Ours-PFS.

D.1 Plan-Solve

The input and output prompts for Plan-Solve on OpenBookQA are shown below.

OpenBookQA: Plan-Solve – Model Input Prompt Example

Select the single best option (A, B, C, D). 'None' is not allowed.

Question: Camouflage can be used by animals for hunting

Options: A. water, B. trees, C. air, D. meals

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

OpenBookQA: Plan-Solve – Model Output Prompt Example

To determine the best option related to camouflage being used by animals for hunting, we need to consider how camouflage functions in the context of hunting.

1. **Water**: While some animals may use water as part of their environment, it is not directly related to camouflage for hunting.
 2. **Trees**: Many animals use trees and foliage to blend into their surroundings, making it easier to ambush prey. This is a strong candidate.
 3. **Air**: While some animals may use the air (like birds) to hunt, it does not directly relate to camouflage in the same way as physical environments do.
 4. **Meals**: This option refers to the food itself rather than the environment used for camouflage.
- Given this analysis, the best option that relates to camouflage being used by animals for hunting is:

B. trees ✗ (Correct: D. meals)

This option directly connects to the idea of animals blending into their environment (like trees) to hunt effectively.

D.2 Ours-PFS (Principles from Failures and Successes)

The input and output prompts for Ours-PFS on OpenBookQA are provided below.

OpenBookQA: Ours-PFS – Model Input Prompt Example

Re-evaluate the following question and your previous answer, applying the relevant principles where applicable to adjust your reasoning. Select the single best option (A, B, C, D). 'None' is not allowed.

Question: Camouflage can be used by animals for hunting

Options: A. water, B. trees, C. air, D. meals

Your Previous Answer: To determine the best option related to camouflage being used by animals for hunting, we need to consider how camouflage functions in the context of hunting.

1. **Water**: While some animals may use water as part of their environment, it is not directly related to camouflage for hunting.
 2. **Trees**: Many animals use trees and foliage to blend into their surroundings, making it easier to ambush prey. This is a strong candidate.
 3. **Air**: While some animals may use the air (like birds) to hunt, it does not directly relate to camouflage in the same way as physical environments do.
 4. **Meals**: This option refers to the food itself rather than the environment used for camouflage.
- Given this analysis, the best option that relates to camouflage being used by animals for hunting is:

B. trees ✗ (Correct: D. meals)

This option directly connects to the idea of animals blending into their environment (like trees) to hunt effectively.

Principles for Avoiding Mistakes:

(F-1) A camouflaged creature in a patterned room will be difficult to see, making it "rough to spot" due to its ability to blend in with the environment. To avoid similar mistakes, always ensure that your reasoning aligns with the fundamental concepts being tested, such as the definition of camouflage in this case.

Principles for Achieving Correct Answers:

⟨S-1⟩ Coloration in animals primarily serves the function of protection, such as through camouflage or warning signals against predators. To solve similar problems, evaluate each option in relation to the context of the question and identify the most relevant function or purpose.

Answer: Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

OpenBookQA: Ours-PFS – Model Output Prompt Example

Let's re-evaluate the question and the options provided, applying the relevant principles to ensure a thorough analysis.

****Question:**** Camouflage can be used by animals for hunting.

****Options:**** A. water B. trees C. air D. meals

****Analysis of Each Option:****

A. ****Water**** - While some animals may hunt in or around water, water itself does not serve as a camouflage mechanism. It is more of a habitat than a camouflage strategy.

B. ****Trees**** - This option is highly relevant. Many animals, such as certain birds and insects, use trees as a backdrop to blend in with their environment while hunting. This directly relates to the concept of camouflage.

C. ****Air**** - Similar to water, air is not a camouflage medium. While some animals may use the air for hunting (like birds), it does not pertain to camouflage in the same way that physical structures do.

D. ****Meals**** - This option refers to the food that animals hunt, but it does not relate to camouflage. Meals are the target of hunting, not a method or environment for camouflage.

****Conclusion:**** After reevaluating the options, the best choice remains: ****B. trees****

This option directly connects to the use of camouflage in hunting, as many animals utilize trees to conceal themselves from their prey.

****Principles for Avoiding Mistakes:**** - Always ensure that the reasoning aligns with the fundamental concepts being tested, such as the definition and application of camouflage in this case.

****Principles for Achieving Correct Answers:**** - Evaluate each option in relation to the context of the question and identify the most relevant function or purpose, which in this case is how camouflage aids in hunting. Thus, the final answer is ****B. trees****.

****Conclusion:**** By applying the principles of understanding the core function, eliminating irrelevant options, and selecting the most relevant option, the most reasonable choice is "B. Trees." ❌ (Correct: D. meals)