

# Training-free LLM Verification via Recycling Few-shot Examples

Dongseok Lee<sup>1</sup> Jimyung Hong<sup>1</sup> Dongyoung Kim<sup>2</sup> Jaehyung Kim<sup>1</sup>

<sup>1</sup>Yonsei University <sup>2</sup>KAIST

{ehdtjr1220, jaehyungk}@yonsei.ac.kr

## Abstract

Although large language models (LLMs) have achieved remarkable performance, the inherent stochasticity of their reasoning processes and varying conclusions present significant challenges. Majority voting or Best-of-N with external verifiers has been explored to mitigate this, but these approaches are limited in applicability or require additional training. To address this, we propose a novel framework that **Recycles Few-shot** examples to **verify** LLM outputs (ReFeri). Our key idea is to utilize the given few-shot examples not only to generate outputs, but also to evaluate the candidate outputs. Specifically, ReFeri combines a *forward confidence score* with a *backward reconstruction penalty* to select candidates that follow few-shot guidance while avoiding demonstration-specific overfitting. Experiments with three different LLMs across seven diverse tasks demonstrate that our framework significantly improves average accuracy of LLMs—achieving an average relative gain of 8.2%—through effective response selection.<sup>1</sup>

## 1 Introduction

Recently, large language models (LLMs) have shown remarkable performance on complex reasoning tasks, such as math, coding, and robotics (Anthropic, 2024; Dubey et al., 2024; OpenAI, 2024c; Team et al., 2023). To further enhance the reasoning capacity of LLMs, various approaches have been proposed, ranging from in-context learning at test time (Wei et al., 2022; Kojima et al., 2022) to recent reinforcement learning methods (Qu et al., 2024; Guo et al., 2025). However, the stochastic nature of LLMs still poses challenges, since different reasoning paths with varying conclusions can be generated for the same input, making it difficult to identify the optimal response. Majority voting approaches (Wang et al., 2023b; Aggarwal

et al., 2023) have been widely adopted to mitigate such randomness by aggregating multiple LLM outputs. However, these approaches are applicable only when the answer can be easily extracted and aggregated. Consequently, they are difficult to apply to open-ended text generation tasks, such as code generation and personalized dialogue generation (Jain et al., 2025; Salemi et al., 2024).

To address this challenge, selecting the most promising solution among multiple LLM outputs, often termed *Best-of-N*, has gained attention (Snell et al., 2024; Gui et al., 2024). Training external verifiers (Cobbe et al., 2021; Lightman et al., 2024) is a representative approach, but it necessitates extensive task-specific labeled data and training. Consequently, *training-free* alternatives, such as the *LLM-as-Judge* framework (Zheng et al., 2023), or recent methods leveraging the model’s internal confidence (Kang et al., 2025; Wang and Zhou, 2024), have been widely explored. However, these methods rely on the model’s *intrinsic knowledge* encoded within the parameters, which can be miscalibrated or unreliable when the model encounters a new domain or complex reasoning (Yuan et al., 2024; Zhang et al., 2025). Essentially, asking an LLM to verify its own output in an unfamiliar domain without external guidance is prone to failure.

Motivated by this limitation, we suggest a new perspective: *utilizing few-shot examples to verify and select among multiple LLM outputs*. Although few-shot examples are often omitted in recent instruction-tuned LLMs due to their zero-shot capabilities (Guo et al., 2025; Sprague et al., 2025), we argue that they remain the most direct and effective medium for conveying human priors. Even if an LLM has not encountered a specific task during training, few-shot examples serve as explicit grounding signals that guide the model on the desired reasoning logic and output format.

In this work, we propose **ReFeri**, a novel framework that **Recycles Few-shot** examples to **verify**

<sup>1</sup>Code: [github.com/dongseok1220/Referi](https://github.com/dongseok1220/Referi).

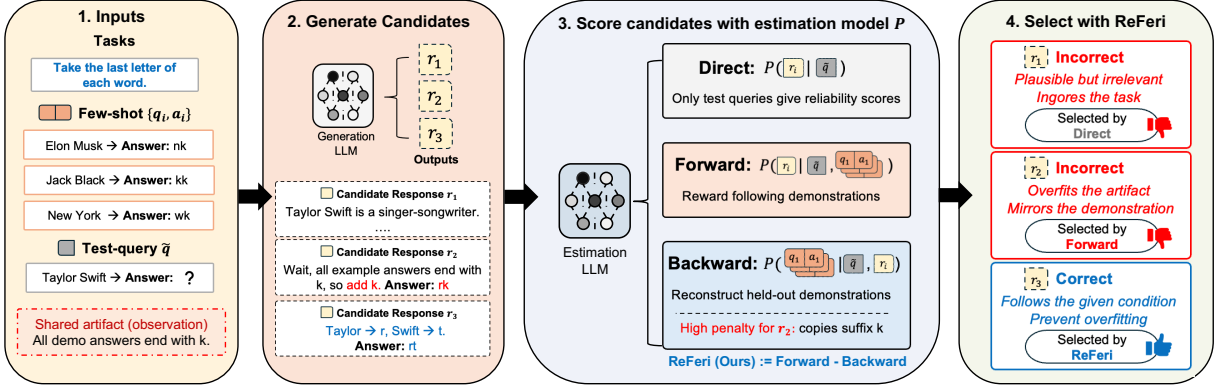


Figure 1: **An overview of ReFeri.** For  $N$  candidate responses from LLMs, ReFeri evaluates each candidate by combining a forward confidence score (*how well the candidate follows the few-shot examples*) and a backward reconstruction penalty (*that penalizes candidates that mimic the surface patterns or artifacts of few-shot examples*). This example illustrates how ReFeri regularizes selection when forward/direct fail.

LLM outputs. The core idea of ReFeri is to utilize the few-shot examples not only to generate answers but also to rigorously evaluate them. To this end, ReFeri exploits the bidirectional relationship between few-shot examples and the target query via a *Bayes-inspired* decomposition. Specifically, ReFeri estimates the quality of each candidate using two complementary signals: (i) *forward confidence score* that measures the likelihood of the candidate given the few-shot examples, ensuring the output follows the provided guidance; and (ii) *backward reconstruction penalty* that suppresses candidates that are *overly specific* to the few-shot demonstrations. Intuitively, a large backward penalty implies that the candidate mimics the surface patterns or artifacts (*i.e.*, overfitting) of the few-shot examples, rather than reflecting the underlying reasoning logic. Consequently, ReFeri enables robust selection without additional model training, effectively balancing the guidance of few-shot examples with the model’s generalization capability.

We validate the effectiveness of ReFeri across three different LLMs (GPT-4o, GPT-4o-mini, and LLaMA-3.1-8B) and seven different benchmarks. When selecting one response among five candidates generated by few-shot chain-of-thought prompting, ReFeri generally outperforms other training-free methods across all tasks, with an average relative gain of 8.2% over random selection and 2.6% over the strongest previous baseline. Importantly, these gains persist even with a lightweight likelihood estimator, yielding a favorable cost–accuracy trade-off and indicating that ReFeri’s effectiveness stems from its Bayes-inspired scoring design rather than estimator capacity alone. We further show that ReFeri scales reliably with the number of candi-

date responses, remains robust across estimation models, and generalizes well to open-ended generation tasks. We also empirically validate the role of the backward reconstruction penalty in capturing demonstration-specific dependence.

## 2 Method

### 2.1 Problem formulation

Let us denote a given LLM as  $\mathcal{M}$  and a target test query as  $\tilde{q}$ . We assume access to a set of  $K$  few-shot examples  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^K$ ,  $\mathbf{x}_i = (q_i, a_i)$ , where each pair consists of a query  $q_i$  and its corresponding ground-truth answer  $a_i$ . Then, *few-shot prompting* incorporates the few-shot examples  $\mathbf{X}$  as the context to guide  $\mathcal{M}$  in generating a response:

$$r_n \sim \mathcal{M}(\tilde{q}, \mathbf{X}), \quad (1)$$

where multiple non-identical predictions  $r_n$ ,  $n = 1, \dots, N$  can be sampled. Then, our goal is to identify the optimal response  $r_{n^*}$  among  $N$  candidates. Existing approaches typically employ a scoring function or decision rule for this selection. For example, Self-Consistency (Wang et al., 2023b) aggregates outputs via majority voting, assuming the answer is unique and strictly extractable. Alternatively, Best-of-N uses an external verifier  $R_\phi$  such as reward models (Cobbe et al., 2021; Lightman et al., 2024) to score the predictions and select the highest-scoring candidate:

$$r_{n^*} = \arg \max_{n=1, \dots, N} R_\phi(\mathbf{y}_n), \quad (2)$$

where  $\mathbf{y}_n = (\tilde{q}, r_n)$ . However, these approaches face significant challenges: voting is ill-suited

for open-ended generation, while training a task-specific verifier requires expensive labeled data and often fails to generalize to unseen domains.

## 2.2 ReFeri: Recycling few-shot examples for training-free LLM verification

Let  $P$  be an estimation model (e.g., a pre-trained LLM) capable of computing the likelihood  $P(\mathbf{y}_n) = P(r_n | \tilde{q})$  of a response  $r_n$  given a query  $\tilde{q}$ .<sup>2</sup> Then, our goal is to identify the best response  $r_{n^*}$  that maximizes this likelihood:

$$n^* = \arg \max_{n=1, \dots, N} P(\mathbf{y}_n). \quad (3)$$

We note that likelihood-based selection has shown effectiveness in finding a high-quality reasoning path (Wang and Zhou, 2024). However, directly estimating  $P(\mathbf{y}_n)$  is not always effective, as it relies entirely on the model’s intrinsic knowledge, which may be uncalibrated in unfamiliar domains (see our empirical results in Appendix B.11). Furthermore, when there is a mismatch between  $\mathcal{M}$  and  $P$ , the estimated likelihoods can be unreliable. To address this, we reinterpret  $P(\mathbf{y}_n)$  with few-shot examples  $\mathbf{X}$ , through Bayes’ rule:

$$P(\mathbf{y}_n) = \frac{P(\mathbf{y}_n | \mathbf{X}) \cdot P(\mathbf{X})}{P(\mathbf{X} | \mathbf{y}_n)}. \quad (4)$$

Then, in log-space, this yields a decomposition into two intuitive forward and backward terms:

$$\underbrace{\log P(\mathbf{y}_n)}_{\text{direct}} = \underbrace{\log P(\mathbf{y}_n | \mathbf{X})}_{\text{forward}} - \underbrace{(\log P(\mathbf{X} | \mathbf{y}_n) - \log P(\mathbf{X}))}_{\text{backward}}. \quad (5)$$

While Eq. 5 represents a mathematical identity, we leverage this decomposition as a strategic design lens to overcome the aforementioned reliability issues. Instead of depending on the model’s potentially uncalibrated confidence, this approach anchors the verification process in the explicit few-shot examples. Specifically, ReFeri substitutes the uncertain direct likelihood with two signals: a *forward confidence score* and a *backward reconstruction penalty*, which we define below.

**Forward confidence score.** Intuitively, the confidence of candidate response  $r_n$  to test query  $\tilde{q}$  is captured by  $\log P(\mathbf{y}_n | \mathbf{X})$ ; this score is high when  $r_n$  aligns well with the reasoning patterns in the

few-shot examples  $\mathbf{X}$ . This forward score has certain advantages over direct estimation of  $P(\mathbf{y}_n)$ , as it allows the estimation to be grounded in the few-shot examples and hence reduces the reliance on its prior knowledge alone. As a result, the forward score provides a more context-aware and robust estimation, especially important in unfamiliar or domain-shifted scenarios. In addition, when the estimation model  $P$  is equal to generation LLM  $\mathcal{M}$ , the forward score can be freely obtained during generation of  $r_n$ . Formally, under the autoregressive LM  $P$ , the forward score is derived as:

$$\begin{aligned} S_{\text{Forw}}(\mathbf{y}_n) &:= \log P(\mathbf{y}_n | \mathbf{X}) \\ &= \frac{1}{L} \sum_{t=1}^L \log P(r_{n,t} | \tilde{q}, \mathbf{X}, r_{n,<t}), \end{aligned} \quad (6)$$

where each candidate response is a sequence of  $L$  tokens  $r_n = (r_{n,1}, \dots, r_{n,L})$ .

**Backward reconstruction penalty.** The backward term  $\log P(\mathbf{X} | \mathbf{y}_n) - \log P(\mathbf{X})$  measures how much conditioning on the candidate pair  $\mathbf{y}_n = (\tilde{q}, r_n)$  increases the likelihood of reconstructing the few-shot examples. A large backward term indicates that the candidate is overly tailored to the few-shot examples, capturing demonstration-specific patterns rather than reasoning specific to the target query; hence, the backward term acts as a penalty. We note that a large backward term may also reflect reasoning patterns shared with the demonstrations (see Section 3.3 for further analysis). Under the assumption of mutual independence between few-shot examples, the backward term can also be derived similarly to Eq. 6. To better utilize given few-shot examples, we refine the backward term using a leave-one-out strategy through prompt replacement; we construct new demonstration  $\tilde{\mathbf{X}}_i$  by replacing  $i$ -th example  $\mathbf{x}_i = (q_i, a_i)$  with  $\mathbf{y}_n$ :

$$\tilde{\mathbf{X}}_i := \mathbf{X}_{-i} \cup \{\mathbf{y}_n\}, \quad (7)$$

where  $\mathbf{X}_{-i}$  denotes the few-shot examples excluding  $\mathbf{x}_i$ . Then, by including  $\tilde{\mathbf{X}}_i$  during estimation for  $\mathbf{x}_i$  as additional input context similar to the forward term, we define the modified backward term:

$$\begin{aligned} S_{\text{Back}}(\mathbf{y}_n) &:= \log P(\mathbf{X} | \mathbf{y}_n) - \log P(\mathbf{X}) \\ &= \frac{1}{K} \sum_{i=1}^K (\log P(a_i | q_i, \tilde{\mathbf{X}}_i) - \log P(a_i | q_i)). \end{aligned} \quad (8)$$

Including the remaining examples  $\mathbf{X}_{-i}$  enables more accurate likelihood estimation for target example  $\mathbf{x}_i$  by leveraging the in-context learning capability of  $P$  (see Appendix B.10). In addition,

<sup>2</sup> $P$  can be different from the generation model  $\mathcal{M}$ ; if  $\mathcal{M}$  is a black-box LLM, using white-box  $P$  is necessary.

similar to Eq. 6,  $\log P(a_i|q_i)$  and  $\log P(a_i|q_i, \tilde{\mathbf{x}}_i)$  can be computed via token-level decomposition.

While Eq. 8 is theoretically rigorous, computing it for all  $K$  examples is expensive. More importantly, we argue that the backward signal is dominated by the examples most semantically related to the query, while distant examples contribute negligible or noisy signals. Therefore, we propose a *nearest-neighbor approximation* that focuses verification on the local semantic neighborhood. Specifically, we identify the single most relevant example  $\mathbf{x}_{i^\dagger}$  using a pre-trained embedding model  $E$ :

$$i^\dagger = \arg \max_{i=1, \dots, K} \cos(E(\tilde{q}), E(q_i)). \quad (9)$$

Then, we define the approximated backward term:

$$\tilde{S}_{\text{Back}}(\mathbf{y}_n) := \log P(a_{i^\dagger} | q_{i^\dagger}, \tilde{\mathbf{x}}_{i^\dagger}) - \log P(a_{i^\dagger} | q_{i^\dagger}) \quad (10)$$

We emphasize that the backward term is not intended as a standalone measure of response quality; rather, it is interpreted jointly with the forward score through their difference (see Section 3.3).

**Final score.** Finally, motivated by Eq. 5, we design our main selection score  $S_{\text{Fin}}$  by subtracting the backward penalty from the forward confidence score to find the most promising output  $r_{n^*}$ :

$$\begin{aligned} n^* &= \arg \max_{n=1, \dots, N} S_{\text{Fin}}(\mathbf{y}_n), \\ S_{\text{Fin}}(\mathbf{y}_n) &:= S_{\text{Forw}}(\mathbf{y}_n) - \tilde{S}_{\text{Back}}(\mathbf{y}_n). \end{aligned} \quad (11)$$

The overall algorithm is presented in Algorithm 1.

## 3 Experiments

### 3.1 Setups

**Datasets.** We evaluate our method on seven benchmarks encompassing diverse reasoning paradigms. (1) *MATH500* (Lightman et al., 2024); a 500-problem subset of MATH (Hendrycks et al., 2021b). (2) *MMLU-Pro* (Wang et al., 2024c); 4,200 examples, including 300 randomly sampled questions per domain. (3) *HotpotQA* (Yang et al., 2018); 500 samples from (Kim et al., 2024), a multi-hop question-answering benchmark. (4) *DROP* (Dua et al., 2019); 500 randomly sampled questions from this reading comprehension benchmark. (5) *GPQA-diamond* (Rein et al., 2024) (*GPQA*); 198 graduate-level questions for complex reasoning. (6,7) *MuSR* (Sprague et al., 2024); 256 examples in Object Placement (*MuSR-op*) and 250 in Team Allocation (*MuSR-ta*) assessing spatial and relational reasoning. Further details are provided in Appendix A.1.

**Baselines.** We compare ReFeri against six widely-used methods that require no additional training, with some reflecting different uses of few-shot examples: (1) *Zero-shot CoT* (Kojima et al., 2022) appends a trigger phrase ("Let's think step by step.") to each query without exemplars. (2) *Few-shot CoT* (Brown et al., 2020) prepends a fixed set of few examples. (3) *LEAP* (Zhang et al., 2024) improves few-shot prompting by inducing mistakes on few examples, extracts task-specific principles through self-reflection, and applies these principles to unseen questions. (4) *USC* (Chen et al., 2023) asks LLM to select the consistent answer from responses. (5) *CoT-WP* (Wang and Zhou, 2024) scores each candidate by the top-1/top-2 token-probability gap at answer positions. (6) *Self-Certainty* (Kang et al., 2025) measures the KL-divergence between the model's next token distribution and the uniform distribution as a selection metric. See Appendix A.2.

**Implementation details.** For the experiments, we use three target LLMs for response generation: gpt-4o-2024-08-06 (*GPT-4o*) (OpenAI, 2024b), gpt-4o-mini-2024-07-18 (*GPT-4o-mini*) (OpenAI, 2024a), and LLaMA-3.1-8B-Instruct (*LLaMA-3.1-8B*) (Dubey et al., 2024). We generate  $N=5$  responses per query using a temperature of 1.0 to encourage diverse candidates. For Zero-shot CoT, Few-shot CoT and LEAP, we report the average accuracy across five, which can be viewed as randomly selecting the response. For USC, CoT-WP, Self-Certainty, and ReFeri, we use the same candidates generated from Few-shot CoT and employ LLaMA-3.1-8B as the estimation (or LLM-judge) model. For computing similarity in the backward term (Eq. 9), we employ all-mpnet-base-v2, a lightweight embedding model with 110M parameters. See Appendix A.3.

### 3.2 Main results

Table 1 summarizes the experimental results on seven reasoning benchmarks across three LLMs. We report results averaged across three independent runs for all methods to mitigate the probabilistic nature of LLM sampling. Across all LLMs and benchmarks, ReFeri improves the average accuracy by 8.2% relative to Few-shot CoT, which corresponds to random selection. Compared to the strongest baseline, Self-Certainty, ReFeri achieves an average relative improvement of 2.6%. To verify that these gains are not an artifact of sampling vari-

Table 1: **Main results.** Overall performance on seven reasoning benchmarks comparing the proposed **ReFeri** with training-free baselines under three generation LLMs. Unless otherwise specified, ReFeri uses LLaMA-3.1-8B as the estimation model; ReFeri<sub>1B</sub> denotes the same method with a LLaMA-3.2-1B estimator. The best and second-best scores within each generation model are highlighted in **bold** and underlined, respectively.

| Models       | Methods              | MuSR-ta<br>(Acc.)               | MuSR-op<br>(Acc.)               | GPQA<br>(Acc.)                  | MATH500<br>(Acc.)               | DROP<br>(EM / F1)                                          | HotpotQA<br>(EM / F1)                                      | MMLU-PRO<br>(Acc.)              | Avg.                            |
|--------------|----------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|------------------------------------------------------------|------------------------------------------------------------|---------------------------------|---------------------------------|
| GPT-4o       | Zero-shot CoT        | 67.5 $\pm$ 0.8                  | 62.1 $\pm$ 0.4                  | 49.5 $\pm$ 0.8                  | 77.1 $\pm$ 0.6                  | 74.2 $\pm$ 0.8 / 84.9 $\pm$ 0.4                            | 37.8 $\pm$ 0.3 / 50.3 $\pm$ 0.4                            | 74.0 $\pm$ 0.2                  | 63.2 $\pm$ 0.2                  |
|              | Few-shot CoT         | 87.2 $\pm$ 0.6                  | 69.6 $\pm$ 0.6                  | 47.3 $\pm$ 1.7                  | 75.5 $\pm$ 0.1                  | 80.4 $\pm$ 0.2 / 89.0 $\pm$ 0.2                            | 44.9 $\pm$ 0.4 / 58.6 $\pm$ 0.3                            | 73.7 $\pm$ 0.2                  | 68.4 $\pm$ 0.3                  |
|              | LEAP                 | 88.1 $\pm$ 1.6                  | 68.0 $\pm$ 1.2                  | 47.8 $\pm$ 2.8                  | 75.2 $\pm$ 0.4                  | 81.0 $\pm$ 0.5 / 89.4 $\pm$ 0.4                            | 44.5 $\pm$ 0.6 / 57.8 $\pm$ 0.5                            | 73.9 $\pm$ 0.2                  | 68.4 $\pm$ 0.4                  |
|              | USC                  | 85.9 $\pm$ 1.9                  | 71.5 $\pm$ 0.7                  | 48.2 $\pm$ 1.6                  | 77.3 $\pm$ 0.3                  | 81.8 $\pm$ 0.3 / 90.2 $\pm$ 0.1                            | 45.9 $\pm$ 0.3 / 60.2 $\pm$ 0.5                            | 74.9 $\pm$ 0.5                  | 69.4 $\pm$ 0.5                  |
|              | CoT-WP               | 88.3 $\pm$ 0.2                  | 67.5 $\pm$ 1.4                  | 49.5 $\pm$ 1.8                  | 78.1 $\pm$ 0.6                  | 83.3 $\pm$ 0.1 / 90.5 $\pm$ 0.8                            | 46.3 $\pm$ 0.9 / 59.6 $\pm$ 0.9                            | 74.4 $\pm$ 0.3                  | 69.6 $\pm$ 0.0                  |
|              | Self-Certainty       | 88.9 $\pm$ 0.5                  | 71.1 $\pm$ 2.1                  | 50.2 $\pm$ 0.8                  | 77.7 $\pm$ 0.4                  | 81.1 $\pm$ 0.8 / 89.4 $\pm$ 0.3                            | 43.9 $\pm$ 0.4 / 57.7 $\pm$ 0.1                            | 74.6 $\pm$ 0.4                  | 69.6 $\pm$ 0.4                  |
|              | ReFeri               | <b>90.8 <math>\pm</math>0.4</b> | <b>73.7 <math>\pm</math>2.2</b> | <b>51.8 <math>\pm</math>0.3</b> | <b>78.5 <math>\pm</math>0.6</b> | <b>83.6 <math>\pm</math>0.2 / 90.9 <math>\pm</math>0.6</b> | <b>46.9 <math>\pm</math>0.4 / 61.0 <math>\pm</math>0.7</b> | <b>75.4 <math>\pm</math>0.3</b> | <b>71.5 <math>\pm</math>0.4</b> |
|              | ReFeri <sub>1B</sub> | 90.7 $\pm$ 0.2                  | <u>74.1 <math>\pm</math>3.8</u> | <u>51.0 <math>\pm</math>1.3</u> | <u>78.3 <math>\pm</math>0.6</u> | <u>83.3 <math>\pm</math>0.5 / 90.7 <math>\pm</math>0.1</u> | <u>46.5 <math>\pm</math>0.8 / 60.5 <math>\pm</math>0.2</u> | <u>75.1 <math>\pm</math>0.1</u> | <u>71.3 <math>\pm</math>0.4</u> |
| GPT-4o-mini  | Zero-shot CoT        | 57.6 $\pm$ 1.3                  | 58.9 $\pm$ 2.7                  | 41.8 $\pm$ 1.0                  | 75.8 $\pm$ 0.6                  | 77.2 $\pm$ 0.7 / <b>85.1 <math>\pm</math>0.8</b>           | 31.5 $\pm$ 0.4 / 41.6 $\pm$ 0.4                            | 63.0 $\pm$ 0.1                  | 58.0 $\pm$ 0.1                  |
|              | Few-shot CoT         | 77.5 $\pm$ 0.5                  | 60.3 $\pm$ 0.8                  | 42.4 $\pm$ 1.1                  | 74.7 $\pm$ 0.7                  | 76.5 $\pm$ 0.2 / 82.9 $\pm$ 0.2                            | 33.6 $\pm$ 0.3 / 44.9 $\pm$ 0.2                            | 63.0 $\pm$ 0.2                  | 61.1 $\pm$ 0.2                  |
|              | LEAP                 | 74.9 $\pm$ 3.2                  | 60.3 $\pm$ 3.2                  | 43.6 $\pm$ 0.3                  | 74.5 $\pm$ 0.1                  | 75.4 $\pm$ 0.4 / 82.5 $\pm$ 0.5                            | 33.3 $\pm$ 0.6 / 44.5 $\pm$ 0.5                            | 63.1 $\pm$ 0.2                  | 60.7 $\pm$ 0.1                  |
|              | USC                  | 76.5 $\pm$ 2.5                  | 60.5 $\pm$ 1.0                  | <u>44.6 <math>\pm</math>1.2</u> | 76.4 $\pm$ 1.2                  | <u>78.8 <math>\pm</math>1.7 / 85.0 <math>\pm</math>1.0</u> | <u>35.1 <math>\pm</math>0.2 / 47.0 <math>\pm</math>0.3</u> | 64.2 $\pm$ 0.5                  | 62.3 $\pm$ 0.5                  |
|              | CoT-WP               | 79.3 $\pm$ 1.3                  | 58.5 $\pm$ 2.6                  | 41.9 $\pm$ 0.9                  | 77.0 $\pm$ 0.7                  | 76.9 $\pm$ 0.6 / 82.7 $\pm$ 0.6                            | 34.7 $\pm$ 0.9 / 45.9 $\pm$ 0.9                            | <u>64.6 <math>\pm</math>0.4</u> | 61.8 $\pm$ 0.5                  |
|              | Self-Certainty       | 81.7 $\pm$ 1.6                  | 60.1 $\pm$ 0.6                  | 41.6 $\pm$ 2.8                  | 76.8 $\pm$ 1.0                  | 76.7 $\pm$ 0.4 / 83.1 $\pm$ 0.6                            | 34.7 $\pm$ 0.1 / 45.9 $\pm$ 0.4                            | 63.9 $\pm$ 0.8                  | 62.2 $\pm$ 0.1                  |
|              | ReFeri               | <b>83.1 <math>\pm</math>0.2</b> | <u>62.0 <math>\pm</math>0.6</u> | <u>44.6 <math>\pm</math>2.6</u> | <b>78.2 <math>\pm</math>0.4</b> | <b>79.1 <math>\pm</math>0.5 / 84.7 <math>\pm</math>0.8</b> | <b>35.7 <math>\pm</math>0.4 / 47.4 <math>\pm</math>0.6</b> | <b>64.9 <math>\pm</math>0.4</b> | <b>63.9 <math>\pm</math>0.5</b> |
|              | ReFeri <sub>1B</sub> | <u>82.8 <math>\pm</math>0.4</u> | <b>62.9 <math>\pm</math>1.4</b> | <b>45.3 <math>\pm</math>1.2</b> | <u>77.7 <math>\pm</math>0.2</u> | <u>78.3 <math>\pm</math>0.8 / 83.9 <math>\pm</math>1.0</u> | <u>35.1 <math>\pm</math>0.5 / 46.6 <math>\pm</math>0.5</u> | <u>64.6 <math>\pm</math>0.5</u> | <u>63.8 <math>\pm</math>0.2</u> |
| LLaMA-3.1-8B | Zero-shot CoT        | 43.3 $\pm$ 1.5                  | 51.9 $\pm$ 1.2                  | 20.9 $\pm$ 0.7                  | 42.9 $\pm$ 1.3                  | 59.7 $\pm$ 0.7 / 65.9 $\pm$ 0.5                            | 15.7 $\pm$ 0.6 / 21.7 $\pm$ 0.5                            | 40.1 $\pm$ 0.4                  | 39.2 $\pm$ 0.2                  |
|              | Few-shot CoT         | 64.5 $\pm$ 2.2                  | 53.4 $\pm$ 0.8                  | 23.7 $\pm$ 0.3                  | 42.4 $\pm$ 0.5                  | 61.1 $\pm$ 0.7 / 66.5 $\pm$ 0.9                            | 19.3 $\pm$ 0.3 / 25.4 $\pm$ 0.4                            | 39.2 $\pm$ 0.4                  | 43.4 $\pm$ 0.5                  |
|              | LEAP                 | 64.7 $\pm$ 3.9                  | 53.3 $\pm$ 5.1                  | 26.8 $\pm$ 1.8                  | 41.9 $\pm$ 0.5                  | 57.0 $\pm$ 1.1 / 62.6 $\pm$ 1.3                            | 19.9 $\pm$ 0.2 / 26.7 $\pm$ 0.1                            | 36.5 $\pm$ 0.7                  | 42.9 $\pm$ 1.2                  |
|              | USC                  | 67.7 $\pm$ 4.5                  | 54.8 $\pm$ 2.2                  | 28.1 $\pm$ 3.1                  | 48.5 $\pm$ 1.2                  | 68.6 $\pm$ 0.9 / 74.3 $\pm$ 1.3                            | <u>25.5 <math>\pm</math>1.1 / 33.4 <math>\pm</math>1.0</u> | <b>45.8 <math>\pm</math>0.4</b> | 48.4 $\pm$ 1.1                  |
|              | CoT-WP               | 71.9 $\pm$ 0.6                  | 53.1 $\pm$ 1.4                  | 30.5 $\pm$ 2.5                  | 48.1 $\pm$ 1.7                  | <b>71.0 <math>\pm</math>0.5 / 75.1 <math>\pm</math>0.6</b> | <b>25.7 <math>\pm</math>0.1 / 33.2 <math>\pm</math>0.4</b> | <u>45.4 <math>\pm</math>0.7</u> | 49.4 $\pm$ 0.4                  |
|              | Self-Certainty       | 72.4 $\pm$ 3.6                  | <u>55.9 <math>\pm</math>0.6</u> | 30.8 $\pm$ 2.6                  | <b>50.7 <math>\pm</math>2.2</b> | 70.8 $\pm$ 1.1 / 76.2 $\pm$ 1.0                            | 25.2 $\pm$ 0.5 / 32.9 $\pm$ 1.0                            | 44.5 $\pm$ 0.8                  | 50.0 $\pm$ 1.1                  |
|              | ReFeri               | <u>76.5 <math>\pm</math>3.4</u> | <b>57.5 <math>\pm</math>0.5</b> | <b>33.5 <math>\pm</math>1.7</b> | <u>50.2 <math>\pm</math>1.6</u> | <u>70.9 <math>\pm</math>1.6 / 76.3 <math>\pm</math>0.5</u> | <b>25.7 <math>\pm</math>0.7 / 33.6 <math>\pm</math>0.6</b> | 44.7 $\pm$ 0.5                  | <b>51.3 <math>\pm</math>0.6</b> |
|              | ReFeri <sub>1B</sub> | <b>76.9 <math>\pm</math>3.2</b> | 55.5 $\pm$ 0.8                  | <u>31.8 <math>\pm</math>2.0</u> | 50.1 $\pm$ 1.7                  | <u>70.9 <math>\pm</math>1.4 / 75.8 <math>\pm</math>1.0</u> | 25.3 $\pm$ 0.5 / 32.9 $\pm$ 0.6                            | 44.2 $\pm$ 0.5                  | <u>50.7 <math>\pm</math>0.5</u> |

ation, we conduct two-sided paired  $t$ -tests on the seven-task average over the three matched runs. All comparisons against CoT-WP and Self-Certainty favor ReFeri, with five reaching  $p < .05$  and the remaining one at  $p=.055$ . See Appendix B.1 for all 95% confidence intervals and unadjusted  $p$ -values.

Interestingly, CoT-WP and Self-Certainty provide notable benefits for smaller open-source models (e.g., LLaMA), whose samples are often verbose or stylistically variable; however, their effectiveness diminishes for stronger models such as GPT-4o-mini and GPT-4o. This is likely because candidates from these models are uniformly high-quality and induce very similar predictive distributions, making entropy-based uncertainty insufficient to separate subtle differences. This also suggests confidence-based selection is vulnerable to miscalibration under estimator-generator mismatch. In addition, we observe that the effectiveness of USC varies substantially across tasks and is sensitive to response ordering (see Appendix B.2).

In contrast, ReFeri explicitly incorporates few-shot demonstrations for validation via two complementary terms. Consequently, even where model-internal confidence becomes unreliable due to uniformly confident outputs, ReFeri maintains robustness by grounding verification in the provided

demonstrations. This also explains the ReFeri<sub>1B</sub> result in Table 1: the gain primarily reflects ReFeri’s Bayes-inspired use of few-shot examples rather than estimator capacity, which helps a 1B estimator preserve most of the improvement. It also contributes to the cost efficiency of ReFeri as shown in the cost analysis in Section 3.3.

**Test-time scaling.** To assess whether ReFeri scales effectively with the number of candidate outputs similar to the conventional Best-of- $N$  method, we evaluate performance as the candidate pool grows. Specifically, we test  $N=\{1, 5, 10, 15, 20\}$  candidates on three representative tasks—*MATH500*, *GPQA*, and *MuSR-ta*—using GPT-4o-mini as the generation model under Few-shot CoT. As  $N$  increases, ReFeri yields consistent improvements. While random selection accuracy decreases on MATH500, ReFeri consistently selects higher-quality responses, improving from 75.8% at  $N=1$  to 79.4% at  $N=20$ . On GPQA, ReFeri raises performance from 41.4% at  $N=1$  to 45.5% at  $N=20$ . The largest gain is observed on MuSR-ta, where accuracy increases from 75.6% to 86.0%.

In contrast, the performance of CoT-WP and USC even degrades as the number of candidates increases, suggesting that these methods do not cap-

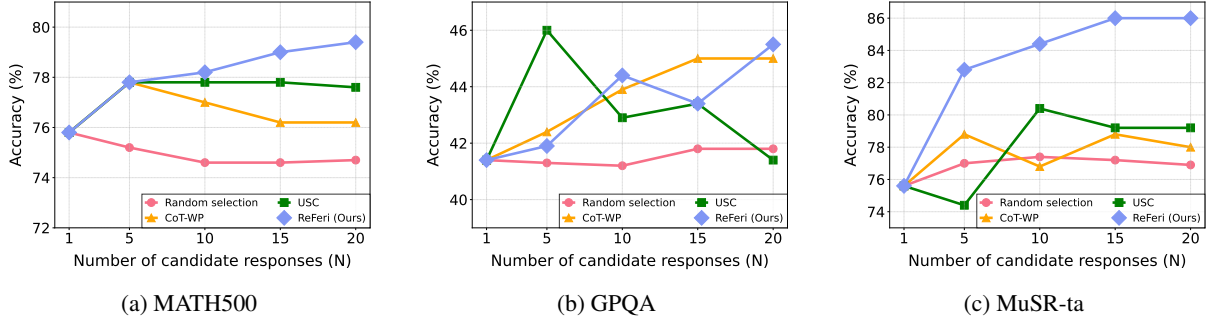


Figure 2: **Test-time scaling.** Accuracy of ReFeri versus other training-free selection methods (Random selection, CoT-WP and USC) on MATH500, GPQA, and MuSR-ta. GPT-4o-mini generates different numbers of candidate responses ( $N = 1, 5, 10, 15, 20$ ) using Few-shot CoT.

Table 2: **Comparison with Self-Consistency.** Seven-task average accuracy under three generation LLMs. Borda weights each vote by the rank induced by the ReFeri score ( $p=1$ ).

| Methods          | LLaMA-3.1-8B | GPT-4o-mini | GPT-4o      |
|------------------|--------------|-------------|-------------|
| Self-Consistency | 50.1         | <b>64.2</b> | 70.7        |
| ReFeri           | <u>51.9</u>  | 63.4        | <u>71.1</u> |
| ReFeri + Borda   | <b>52.4</b>  | <u>64.9</u> | <b>71.3</b> |

ture what is truly plausible among the candidates. Notably, USC demonstrates strong performance on GPQA when  $N=5$ , but its accuracy declines as  $N$  increases, highlighting sensitivity to the candidate set size. Overall, these results confirm that ReFeri scales well with more candidates, demonstrating effectiveness and reliability in test-time scaling.

### 3.3 Additional analyses

In this section, we present additional analyses of ReFeri. For experiments, we use candidate responses from the first run in Table 1 except for Table 3. More results are in Appendix B.

**Comparison with Self-Consistency.** As noted in Section 1, standard *Self-Consistency* (Wang et al., 2023b) requires answers to be normalized into a discrete format for voting, and is therefore inapplicable to open-ended tasks. We therefore use USC as the corresponding baseline in Table 1. Nevertheless, when answer-level voting is feasible, ReFeri can complement Self-Consistency by providing a weight for each candidate’s vote. As shown in Table 2, this weighted voting consistently improves over conventional majority voting across all three generators. These results suggest that ReFeri can serve not only as a standalone response selector, but also as an auxiliary signal for existing voting-based aggregation methods. See Appendix B.3 for further details, including full task-wise accuracy.

Table 3: **Ablation study.** We report a scoring variant averaged over three runs sampled from LLaMA-3.1-8B responses, corresponding to the setup in Table 1. The notations F, B, and D stand for the forward, backward, and direct term, respectively.

|               | F | B | D | MATH500 | GPQA | MuSR-ta |
|---------------|---|---|---|---------|------|---------|
|               | ✗ | ✗ | ✗ | 42.4    | 23.7 | 64.5    |
|               | ✗ | ✗ | ✓ | 50.3    | 32.6 | 68.4    |
|               | ✓ | ✗ | ✗ | 50.1    | 32.3 | 76.9    |
| ReFeri (Ours) | ✓ | ✓ | ✗ | 50.2    | 33.5 | 76.5    |

**Ablation study.** To understand how each component contributes to ReFeri, we conduct an ablation study on the proposed scoring function (Eq. 11). Table 3 shows that the direct score can be effective on relatively familiar tasks like MATH500, where the model’s inherent prior knowledge is often sufficient. However, its performance degrades on complex or less familiar benchmarks (*e.g.*, GPQA and MuSR-ta), demonstrating that directly estimating likelihood loses reliability when the model’s confidence is miscalibrated. In contrast, the forward score exhibits superior performance on MuSR-ta, where few-shot guidance is crucial. Simultaneously, forward score underperforms the direct score on MATH500 and GPQA, suggesting that conditioning on demonstrations is not always beneficial. This is because accidental patterns in few-shot examples are not aligned with valid solutions for a given query, leading to an excessive preference for candidates that closely mimic the demonstrations. However, incorporating the backward reconstruction penalty addresses this problem, as shown in the last row of Table 3, implying the complementary roles of the forward and backward terms. Extending this ablation to all three generators and seven tasks, ReFeri achieves the best task average for every generator, never below both Direct and Forward score, with consistent gains across the three

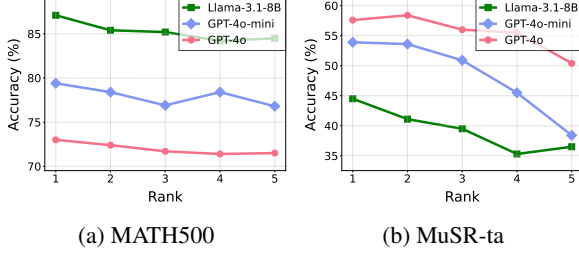


Figure 3: **What the backward term captures.** Candidates are ranked by the raw backward term, where Rank 1 denotes the largest backward term. We measure how well it reconstructs the original few-shot answers.

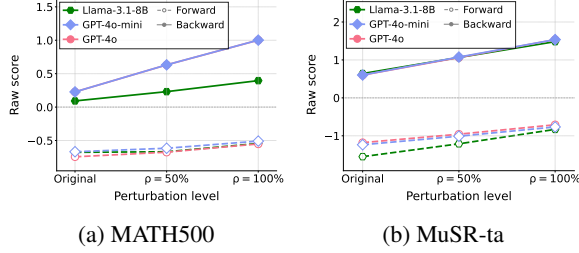


Figure 4: **Perturbation with demonstration fragments.** Few-shot fragments increase the raw backward term more sharply than the forward score.

matched runs ( $p < .05$ ). See Appendix B.10.

**Deeper analysis of the backward term.** To verify that the backward term captures surface-level overfitting to the few-shot examples, we test the following intuition: candidates with larger backward terms should be more effective at recovering the original few-shot answers. Concretely, we rank the generated outputs by the raw backward term and use each output as a one-shot demonstration to isolate the effect of candidate response under the same setup (see Appendix B.10 for the simplified variant). For consistency with our pipeline, we employ LLaMA-3.1-8B as the generation model. As shown in Figure 3, accuracy monotonically decreases with rank: candidates with larger backward terms better reconstruct the original few-shot answers, a behavior that is precisely the failure mode we aim to suppress. In other words, these are candidates that achieve high scores by reproducing demonstration-specific structures rather than by reasoning about the target query. Since the forward score already rewards consistency with the few-shot guidance, subtracting the backward term prevents ReFeri from selecting such mimicry.

To further verify the backward term’s sensitivity to demonstration artifacts, we conduct a perturbation experiment: we prepend the first  $\rho\%$  of the few-shot examples to each candidate response and

Table 4: **Perturbation applied to response selection.** Selection accuracy when the first  $\rho\%$  of the retrieved demonstration answer is prepended to a low-forward incorrect candidate.

| Estimator    | Method        | $\rho=0$ | $\rho=50$    | $\rho=100$   |
|--------------|---------------|----------|--------------|--------------|
| LLaMA-3.1-8B | Forward       | 45.29    | 42.03        | 33.31        |
|              | ReFeri (Ours) | 45.29    | <b>47.48</b> | <b>47.48</b> |
| LLaMA-3.2-1B | Forward       | 45.41    | 41.38        | 30.71        |
|              | ReFeri (Ours) | 45.47    | <b>47.60</b> | <b>47.66</b> |

Table 5: **Matched-pair analysis of the backward term.** Pair accuracy for the highest-forward correct and incorrect candidates in each query, grouped by which is more similar to the retrieved demonstration answer.

| Pair group                           | Forward      | ReFeri (Ours)        |
|--------------------------------------|--------------|----------------------|
| Correct-more-similar ( $n = 316$ )   | <b>74.68</b> | 74.05 (−0.63)        |
| Incorrect-more-similar ( $n = 314$ ) | 44.90        | <b>47.13</b> (+2.23) |

recalculate the forward and backward terms. As shown in Figure 4, inserting demonstration fragments causes a significant increase in the backward term, whereas the change in forward score is relatively small. We next test how this sensitivity affects response selection by perturbing a *low-forward incorrect* candidate. We compare the resulting selection accuracy of Forward and ReFeri across three generators (GPT-4o, GPT-4o-mini, and LLaMA-3.1-8B) and three tasks (MATH500, GPQA, and MuSR-ta). In Table 4, as  $\rho$  increases, forward-only accuracy drops from 45.3% to 33.3%, whereas ReFeri reaches 47.5%, as the injected content incurs a larger backward penalty.

A high backward score can also reflect legitimate reuse of demonstration reasoning. However, this effect is asymmetric (Table 5): when the correct candidate is more demonstration-similar, the backward term reduces pairwise accuracy by only 0.6%, as that candidate retains strong forward support, whereas it yields a 2.2% gain when the incorrect candidate is more similar. Consistently, the raw backward score correlates with candidate–demonstration similarity, but not with correctness ( $|r| \leq 0.09$ ) or response length ( $|r| \leq 0.23$ ). Together with the low agreement between forward and backward selections reported in Table 27 (average agreement of 0.22), these results demonstrate that the backward term provides an orthogonal and mechanistically interpretable signal rather than redundant likelihood information, supporting its role as a backward reconstruction penalty. Further details are provided in Appendix B.12.

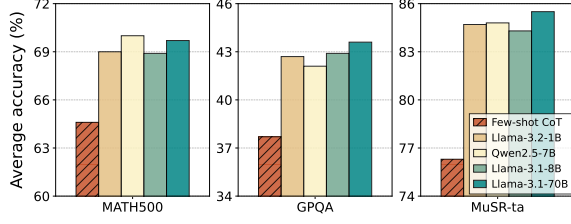


Figure 5: **Estimation models.** Each bar shows the average accuracy of three generation LLMs.

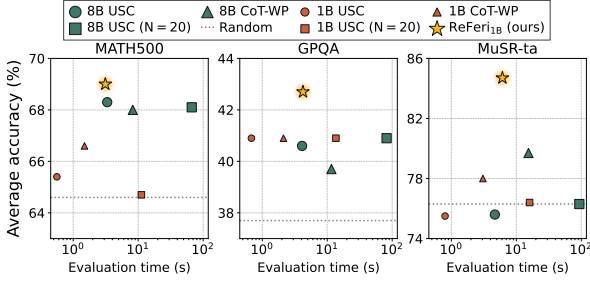


Figure 6: **Cost-accuracy trade-off.** Per-instance evaluation cost (seconds, single NVIDIA RTX A6000 GPU) against accuracy on selected benchmarks.

**Estimation models.** To examine robustness to the choice of estimation model  $p_\theta$ , we evaluate ReFeri’s performance using three LLMs of diverse scales and backbones: LLaMA-3.2-1B-Inst, Qwen-2.5-7B-Inst, and LLaMA-3.1-70B-Inst. The generation model is fixed (either GPT-4o-mini, GPT-4o, or LLaMA-3.1-8B), and the average accuracy is presented in Figure 5. Across all settings, ReFeri consistently improves over Few-shot CoT while the best-performing estimator varies depending on the task; for example, Qwen-2.5-7B leads on MATH500, LLaMA-3.1-70B leads on the more challenging GPQA. This likely reflects differences in how well each model’s intrinsic knowledge aligns with the reasoning style of each benchmark. Despite these task-specific differences, the overall improvements are consistent across all estimation models. This suggests that the effectiveness of ReFeri primarily stems from its validation strategy with few-shot examples, rather than the specific choice of the estimator. More discussion is provided in Appendix B.9.

**Robustness of ReFeri.** Since likelihood-based scoring can be sensitive to context, we examine ReFeri under variations in few-shot examples, prompt format, and estimator calibration. Across independently sampled exemplar sets, ReFeri consistently outperforms Few-shot CoT (Table 14), and the gains persist even when the exemplars are intentionally degraded in reasoning quality (Ta-

Table 6: **LLM personalization.** Evaluation results on LaMP-4 and LaMP-5 using GPT-4o as generator.

| Methods      | LaMP-4       |              | LaMP-5       |              |
|--------------|--------------|--------------|--------------|--------------|
|              | Rouge-1      | Rouge-L      | Rouge-1      | Rouge-L      |
| Vanilla      | 0.126        | 0.112        | 0.424        | 0.342        |
| Few-shot RAG | 0.152        | 0.135        | 0.430        | 0.356        |
| USC          | <u>0.154</u> | 0.136        | 0.437        | 0.366        |
| CoT-WP       | <u>0.154</u> | <u>0.137</u> | <u>0.438</u> | <u>0.368</u> |
| ReFeri       | <b>0.157</b> | <b>0.140</b> | <b>0.451</b> | <b>0.387</b> |

ble 15). ReFeri is also robust to prompt-format variations at both generation and verification (Table 17) and to estimator temperature, which produces negligible changes in performance (Table 18). Finally, we test the robustness of the nearest-neighbor approximation in: even the worst-performing fixed exemplar remains competitive, while increasing the number of retrieved neighbors leaves average performance nearly unchanged (Tables 19 and 20). Together, these results show that ReFeri does not rely on a particular exemplar set, prompt format, calibration choice, or retrieval configuration. Further analyses are provided in Appendix B.5–B.8.

**Computational cost.** Next, we examine the trade-off between cost and accuracy. Since all methods select from the same pre-generated pool of candidates, the generation cost is shared, so not considered. Figure 6 plots accuracy against the average additional evaluation cost for response selection across the three generator models, where points closer to the upper-left indicate a better cost-accuracy trade-off. Full results are provided in Table 22. ReFeri with a 1B estimator lies on a favorable trade-off frontier, reaching 69.0% on MATH500, 42.7% on GPQA, and 84.7% on MuSR-ta, while running approximately  $2.6\times$  faster than CoT-WP with the 8B estimator. In contrast, 1B USC has the lowest computational cost, but its performance remains on par with random selection. Even when USC uses more candidates for voting, its performance actually degrades on benchmarks such as MATH500 (68.3% to 68.1%), despite requiring significantly more computation. CoT-WP also drops significantly from 8B to 1B, showing that other likelihood-based baselines do not retain their effectiveness when scaled down. Overall, combining ReFeri with a small-scale estimator offers significant advantages and provides robust verification at a remarkably low computational cost.

**Generalization to open-ended tasks.** We further apply ReFeri to two open-ended generation

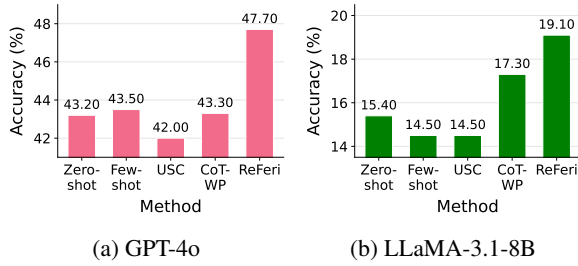


Figure 7: **LiveCodeBench-code generation.** Evaluation results on LiveCodeBench code-generation.

tasks, *LLM personalization* and *code generation*, to evaluate its broader applicability beyond reasoning benchmarks, where majority voting and answer-extraction approaches are inherently limited. Implementation details are provided in Appendix A.3. First, on the *LaMP* benchmark (Salemi et al., 2024), which aims to steer LLM responses towards individual users, we apply ReFeri<sup>3</sup> to LaMP-4 (news headline generation) and LaMP-5 (scholarly title generation) using GPT-4o as the generator. We compare against *Vanilla* (no user context) and *Few-shot RAG*, which augments the prompt with  $K=5$  examples retrieved via BM25 (Robertson and Zaragoza, 2009) from the user’s history. Following (Salemi et al., 2024), we evaluate all responses against gold references using ROUGE-1 and ROUGE-L. As shown in Table 6, ReFeri consistently outperforms all baselines; CoT-WP offers only limited gains on LaMP-4, where short outputs render confidence-based signals uninformative. Next, on *LiveCodeBench* (Jain et al., 2025), which requires both algorithmic correctness and executability, we use GPT-4o and LLaMA-3.1-8B as generation models. As shown in Figure 7, ReFeri achieves the best accuracy: USC is no better than random selection, while CoT-WP struggles especially on GPT-4o, suggesting a calibration mismatch in likelihood-based selection.

## 4 Related Works

**Few-shot in-context learning of LLM.** Few-shot in-context learning (ICL) enables LLMs to generalize to unseen tasks from a handful of demonstrations (Brown et al., 2020). For complex reasoning, Chain-of-Thought (CoT) prompting appends intermediate steps to the few-shot examples (Wei et al., 2022; Fu et al., 2023; Jin et al., 2024). By retrieving better examples, a method has also emerged to further enhance ICL (Wu et al., 2023;

<sup>3</sup>Since personalization relies heavily on retrieved examples, we use the full backward term for rigorous evaluation.

Peng et al., 2024). However, ICL does not always guarantee improvements (Min et al., 2022), and the performance gap between zero-shot and few-shot CoT is narrowing as instruction tuning becomes more effective (Sprague et al., 2025). Nonetheless, carefully selected demonstrations are still effective (Huang et al., 2024), reducing overconfidence in multi-step reasoning (Ge et al., 2025), and mitigating hallucinations and memory-based mistakes in complex tasks (Yan et al., 2025).

**Selection of diverse LLM outputs.** Due to the probabilistic nature of LLM decoding, a variety of outputs are generated, each reflecting different reasoning paths (Kadavath et al., 2022; Qiu and Mikkilineni, 2024). Best-of- $N$  selection aims to identify the most promising candidate among them, but existing methods face several limitations. Self-consistency (Wang et al., 2023b) selects the majority answer, assuming a well-formatted answer that is often violated in open-ended tasks (e.g., summarization or dialogue) (Stiennon et al., 2020; Salemi et al., 2024). Outcome (Cobbe et al., 2021; Uesato et al., 2022), or Process Reward Models (Lightman et al., 2024; Wang et al., 2024b) require domain-specific checkers, limiting scalability to new domains. To eliminate external verifiers, alternatives that require no training leverage the model’s internal signals (e.g., LLM-as-Judge, Self-Certainty) (Chen et al., 2023; Zheng et al., 2023; Kang et al., 2025; Wang and Zhou, 2024), but often lead to miscalibration in unfamiliar domains. In particular, LLM-as-Judge requires fine-tuning to be reliable, reintroducing supervision (Yuan et al., 2024; Mahan et al., 2024; Zhang et al., 2025). In contrast, ReFeri is training-free and task-agnostic, offering a more scalable and generalizable approach by recycling few-shot examples for verification.

## 5 Conclusion

We propose ReFeri, a training-free framework to find promising LLM output. By reusing few-shot data not only for generation but also for verification, ReFeri mitigates confidence miscalibration. In experiments, ReFeri is consistently effective in various LLMs and tasks, while remaining effective with lightweight estimators. Overall, these results suggest that ReFeri is a practical way to find reliable LLM output with minimal human involvement, opening future directions to reconsider the broader utility of few-shot examples.

## Limitations

ReFeri assumes access to few-shot demonstrations. While their choice and quality can affect the candidate pool, ReFeri remains effective across different and deliberately weakened exemplar sets (Appendix B.5), suggesting that carefully curated demonstrations are not essential. Proxy examples from retrieval or stronger models may therefore suffice, though settings with no available demonstrations remain an open direction.

In addition, since the selection by ReFeri is determined by likelihoods produced by an estimation model, it is limited to explain why a response is incorrect, unlike PRMs, which offer step-level feedback, or LLM-as-judge, which can easily generate explanations by prompting. However, we believe that ReFeri can potentially provide a certain level of interpretability; for example, the token-level uncertainty signals derived from the backward-term contributions of candidate responses could potentially reveal untrustworthy tokens. Future work could develop more systematic token-level or step-level diagnostic tools from these decomposed signals, allowing ReFeri to not only select reliable responses but also explain the source of unreliability.

## Broader Impact and Ethical Implications

ReFeri provides a training-free method for selecting promising outputs from LLMs. This makes it particularly valuable in scenarios where labeled data is scarce or where model fine-tuning is impractical such as limited access to data, or applications in emerging domains where predefined labels are unavailable. In addition, ReFeri reduces the barrier to deploying LLMs in real-world settings without additional supervision. This may contribute to broader and more efficient adoption of LLMs in resource-constrained environments. All datasets used are public and widely adopted.

## Acknowledgments

Dongseok Lee and Jaehyung Kim are affiliated with the Department of Artificial Intelligence at Yonsei University. This research was supported in part by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2025-25442405, Development of a Self-Learning

World Model-Based AGI System for Hyperspectral Imaging).

## References

- Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. 2023. Let’s sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Anthropic. 2024. Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*.
- Yuyao Ge, Shenghua Liu, Yiwei Wang, Lingrui Mei, Lizhe Chen, Baolong Bi, and Xueqi Cheng. 2025. Innate reasoning is not enough: In-context learning enhances reasoning large language models with less overthinking. *arXiv preprint arXiv:2503.19602*.
- Lin Gui, Cristina Gârbasea, and Victor Veitch. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Xijie Huang, Li Lyna Zhang, Kwang-Ting Cheng, Fan Yang, and Mao Yang. 2024. Fewer is more: Boosting math reasoning with reinforced context pruning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Live-codebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*.
- Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. In *International Conference on Learning Representations (ICLR)*.
- 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 (NeurIPS)*.
- Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations (ICLR)*.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.
- Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. <https://platform.openai.com/docs/models/gpt-4o-mini>.
- OpenAI. 2024b. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>.
- OpenAI. 2024c. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>.
- Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- 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*.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations (ICLR)*.
- Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations (ICLR)*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Liang Wang, Nan Yang, and Furu Wei. 2024a. Learning to retrieve in-context examples for large language models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.
- Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024c. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Kai Yan, Yufei Xu, Zhengyin Du, Xuesong Yao, Zheyu Wang, Xiaowen Guo, and Jiecao Chen. 2025. Recitation over reasoning: How cutting-edge language models can fail on elementary school-level reasoning problems? *arXiv preprint arXiv:2504.00509*.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024. In-context principle learning from mistakes. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*.

## A More Details of Experimental Setups

This section covers more details about the experiments from Section 3.

### A.1 Datasets

This subsection provides more information about the dataset and the few-shot examples we used.

- **MATH500.** The MATH benchmark (Hendrycks et al., 2021b) consists of 12,500 LaTeX-formatted competition-level math problems, with topics ranging from algebra and geometry to number theory. Each problem includes a step-by-step solution and expects the model to generate a boxed final answer (e.g., an integer or simplified expression). We use MATH500, a 500-question subset introduced in (Lightman et al., 2024), uniformly sampled from the test split to preserve subject and difficulty distribution. For few-shot examples, we follow 5-shot examples from (Yang et al., 2024)<sup>4</sup> for GPT-based models and 4-shot examples from (Lewkowycz et al., 2022)<sup>5</sup> for LLaMA-based models. The reason for this choice is based on our empirical observation: Simply adding "Please think step by step and put your final answer within `\boxed{}`." as done in GPT-style few-shot prompts led to a significant drop in accuracy. Namely, LLaMA-based models require prompt formats that are aligned with their own instructions and are sensitive to deviations from the learned template. This benchmark evaluates symbolic reasoning ability in mathematical domains.

- **MMLU-Pro.** MMLU-Pro (Wang et al., 2024c) is an extension of the original MMLU benchmark (Hendrycks et al., 2021a), which evaluates broad knowledge and reasoning over 57 subjects using 14k 4-way multiple-choice questions. MMLU-Pro introduces 12k curated 10-way multiple-choice questions across 14 professional domains, increasing task difficulty and emphasizing complex, multi-step reasoning. Instead of using the full test set, we subsample 300 questions per subject (totaling 4,200) using random seed 42 and we will share the used indices at the code. For few-shot examples, we used 5-shot examples following the format used in (Wang et al., 2024c). This benchmark is used

<sup>4</sup><https://github.com/QwenLM/Qwen2.5-Math>

<sup>5</sup><https://huggingface.co/datasets/meta-llama/llama-3.2-3B-Instruct-evals>

to assess domain-specific and robust reasoning performance.

- **GPQA.** GPQA (Rein et al., 2024) is a graduate-level QA benchmark consisting of 448 expert-authored multiple-choice questions in domains such as physics, chemistry, and biology. Designed to be "Google-proof," it focuses on evaluating complex scientific reasoning that cannot be answered through simple retrieval. We evaluate on GPQA-Diamond, a curated subset of 198 especially difficult questions selected by the authors. Few-shot examples are taken directly from the official 5-shot examples used in (Rein et al., 2024). This task measures deep scientific understanding.
- **DROP.** The DROP benchmark (Dua et al., 2019) contains 96k question-answer pairs requiring discrete reasoning over Wikipedia passages (e.g., numerical operations, counting, or date comparison). Answers may include spans, numbers, or dates. We evaluate on a 500-sample subset randomly selected from the dev set, and we will share the selected indices at the code. We use 3-shot examples from (Zhou et al., 2023) and report both EM and F1 metrics following the official implementation. This benchmark evaluates models' symbolic reasoning grounded in natural language passages.
- **HotpotQA.** HotpotQA (Yang et al., 2018) consists of 113k multi-hop QA pairs requiring reasoning over multiple Wikipedia documents. The model must retrieve at least two relevant passages and combine facts to answer each question. We follow the (Kim et al., 2024), which uses 500 samples from the dev set. Few-shot examples are taken from (Yao et al., 2023) using the same 6-shot examples. This task tests compositional reasoning and the ability to aggregate distributed information across documents.
- **MuSR.** MuSR (Sprague et al., 2024) is a benchmark for multi-step reasoning over long-form narratives (800–1000 words), constructed via neuro-symbolic generation to embed logical dependencies into natural language. It includes structured tasks such as TeamAllocation (constraint-based planning) and ObjectPlacement (spatial consistency reasoning). We evaluate on the 250 TeamAllocation and 256 ObjectPlacement examples from the official release (Sprague et al., 2024), using 3-shot examples tailored to each task (Sprague et al., 2025). MuSR requires understanding of narrative flow, contextual logic, and physical feasibility. Notably, prior work (Sprague et al., 2025) has shown that few-shot Chain-of-Thought (CoT) prompting yields significant gains over zero-shot CoT in MuSR, highlighting the role of in-context examples in complex reasoning.
- **LaMP.** The LaMP benchmark (Salemi et al., 2024) evaluates personalized text generation across multiple tasks. We focus on LaMP-4 (personalized news headline generation) and LaMP-5 (personalized scholarly title generation). In LaMP-4, the model generates headlines conditioned on article content and author profile, while LaMP-5 requires generating research paper titles conditioned on researcher profiles. We follow the official test splits and evaluate with ROUGE-L and ROUGE-1. Few-shot RAG is an example of 5 shots retrieved from user history via BM25. These tasks test personalization and style-aware generation in open-ended outputs.
- **LiveCodeBench.** LiveCodeBench (Jain et al., 2025) is an execution-based benchmark for evaluating LLM code-generation ability. We use the `code_generation_lite` test split with `release_v3`. Each instance provides a programming problem specification and, for some problems, starter code. The model must generate an executable Python solution, which is evaluated by running the generated code against the corresponding test cases. This benchmark measures both algorithmic correctness and executability, making it suitable for evaluating open-ended code generation.

## A.2 Baselines

This subsection provides additional details on the baselines used in our experiments. We further include the prompt template used for the baselines, with MATH500 serving as a representative benchmark, in Listings 1–7.

- **Zero-shot CoT.** Zero-shot Chain-of-Thought (Kojima et al., 2022) is a method that induces step-by-step reasoning in a model by adding a simple reasoning prompt to the input query without requiring separate in-context examples. Typically, a phrase such as "Let's think step by step" is appended to the query to ensure that

the model generates intermediate reasoning steps rather than outputting only the final answer. However, since recent instruction-tuned models differ in their preferred prompt formats and instruction-following behaviors, the effectiveness of the same trigger phrase may vary depending on the model. In our experiments, Zero-shot CoT is used as the baseline to evaluate how well models can generate reasoning processes and arrive at the correct answer when provided only with reasoning instructions and no additional examples.

- **Few-shot CoT.** Few-shot Chain-of-Thought (Brown et al., 2020) is an approach that provides several CoT demonstrations alongside a test query to guide the model to follow the reasoning and response formats of the examples. Each demonstration consists of a question, a step-by-step reasoning process, and a final answer. Rather than simply adding examples, the prompt is constructed by combining the reasoning instructions used in Zero-shot CoT with few-shot CoT examples. In other words, the model first learns the problem-solving approach and output format from the provided CoT examples through in-context learning, and then generates a reasoning path and final answer for the test query. Therefore, Few-shot CoT serves as a baseline for evaluating how well a model can generalize reasoning patterns to new problems using only a small number of reasoning examples and can be viewed as a random selection method.
- **LEAP.** Learning Principles (LEAP) (Zhang et al., 2024) is a method that intentionally induces errors in few-shot examples and then extracts task-specific principles through the model’s self-reflection to improve prompting. Specifically, the model first generates multiple reasoning outputs for the few-shot examples and uses the outputs that do not match the correct answer as mistake cases. It then extracts example-specific low-level principles from each mistake case and generalizes them to generate high-level principles applicable to unseen questions. In our experiments, following LEAP’s default settings, we set temperature to 1.0 to generate  $N = 20$  reasoning outputs and add the extracted high-level principles to the few-shot CoT prompt to generate the final answer. This procedure does not require human-written annotations, since the principles are generated by the model itself. This baseline evaluates whether explicit task-level principles can improve reasoning beyond standard few-shot demonstrations.
- **USC.** Universal Self-Consistency (Chen et al., 2023) is a prompt-based selection method that requires no additional training and enables the model to select a consistent answer from among multiple Chain-of-Thought outputs. Standard Self-consistency (Wang et al., 2023b) methods select the most frequent answer from sampled inference paths, implicitly assuming that the majority opinion is more likely to be accurate. However, this approach assumes that the correct answer can be easily extracted and aggregated, and it cannot be applied to open-ended tasks where this is not possible. To address this limitation, USC utilizes the model itself as an evaluator. Specifically, it presents the model with multiple candidate responses and asks it to select the most reliable one.
- **CoT-WP.** Chain-of-Thought Reasoning without Prompting (CoT-WP) (Wang and Zhou, 2024) is a method that scores each candidate response using the token-level probabilities of a language model conditioned on the same few-shot prompt. Specifically, it calculates the probability gap between the top-1 token and the top-2 token at each answer token position that makes up the final answer. When an answer consists of multiple tokens, this experiment calculates the confidence gap for every answer token position and averages these values to obtain the final score for the candidate response. A higher value indicates that the model is more confident in the selected answer token than in the alternative tokens. Finally, the candidate response with the highest CoT-WP score is selected. In our experiments, CoT-WP serves as a baseline closely related to ReFeri because both methods require no training and rely on the model’s intrinsic likelihood signals to evaluate candidate responses.
- **Self-Certainty.** Self-Certainty (Kang et al., 2025) is a method that estimates the reliability of a generated response using the model’s next-token distribution. Specifically, it calculates the KL divergence between the model’s next-token distribution and a uniform distribution. The closer the distribution is to a uniform distribution, the more the model’s prediction is spread across multiple tokens, indicating low cer-

tainty; conversely, the more it is concentrated on a specific token, the higher the certainty. In this experiment, we use this certainty score to evaluate the reliability of candidate responses. Therefore, Self-Certainty is used as a baseline to assess whether the uncertainty signal derived from the model’s internal predictive distribution can identify more reliable generations.

### A.3 Implementation

This section provides the detailed information needed to implement the main experiment.

**Resource details.** To avoid out-of-memory errors, we used two NVIDIA H100 GPUs for evaluation with the LLaMA-3.1-70B-Instruct model. All other experiments were performed on a single NVIDIA RTX A6000 GPU.

**Response generation.** We use lm-eval-harness<sup>6</sup> to generate responses from LLaMA-based models, with temperature set to 1.0 and 5 responses per input. The prompt<sup>7</sup> was written in chat template format using vllm. For GPT-family models, we use the official OpenAI API to generate completions under the same sampling configuration. The remaining settings follow the GPT API default settings. During evaluation, for Zero-shot CoT, Few-shot CoT, and LEAP, we average performance over five sampled generations, which can be viewed as random selection from the candidate pool. In contrast, for USC, CoT-WP, Self-Certainty, and ReFeri, we evaluate the single response selected from the same candidate pool. All evaluations are conducted using our custom evaluation scripts to ensure consistent scoring and formatting across models.

**Algorithm of ReFeri.** In algorithm 1, we present the formal algorithm for ReFeri. We generate multiple candidate responses  $\{r_1, \dots, r_N\}$  for each test query using Few-shot CoT, as it exhibits better quality on the average (see Table 1).

**Application to LLM personalization.** In the personalization experiments, ReFeri is applied to two open-ended generative LaMP tasks: personalized news headline generation (LaMP-4) and personalized academic paper title generation (LaMP-5). For each task, we use a 100-user subset, with 6,725 test queries for news headline generation and 106

---

#### Algorithm 1 ReFeri algorithm

---

**Input:** estimation model  $P$ , embedding model  $E$ , test-query  $\tilde{q}$ ,  $N$  candidate responses  $\{r_1, \dots, r_N\}$ ,  $K$  few-shot examples  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^K$

---

```

 $i^\dagger = \arg \max_{i=1, \dots, K} \cos(E(\tilde{q}), E(q_i))$ 
for  $n = 1$  to  $N$  do
     $S_{\text{Forw}} \leftarrow$  Compute forward score with  $r_n$  as
    label, using  $P$  and context  $(\tilde{q}, \mathbf{X})$  (Eq. 6)
    Construct  $\tilde{\mathbf{X}}_{i^\dagger} \leftarrow$  using a leave-one-out strat-
    egy (Eq. 7)
     $S_{\text{Back}} \leftarrow$  backward score with  $a_{i^\dagger}$  as label,
    using  $P$  and  $\tilde{\mathbf{X}}_{i^\dagger}$  (Eq. 10)
     $S_{\text{Fin}} \leftarrow S_{\text{Forw}} - S_{\text{Back}}$  (Eq. 11)
     $S_n \leftarrow S_{\text{Fin}}$ 
end for
 $r_{n^*} \leftarrow \arg \max_n S_n$ 
return  $r_{n^*}$ 

```

---

test queries for academic title generation. For each query, we use BM25 to search for relevant examples from the user’s history and construct a few-shot personalized prompt using  $K = 5$  of the retrieved examples. We use the full backward term rather than the nearest-neighbor approximation for this experiment, since personalization relies heavily on the retrieved user examples. Additionally, a pre-built natural language user profile is appended to the prompt for each user. GPT-4o is used as the generator, with the temperature set to 1.0, and 5 candidate outputs are sampled for each query. The maximum generation length is limited to 128 tokens. As in Table 1, Llama-3.1-8B-Instruct is used as the likelihood estimator. Following the LaMP evaluation protocol, the generated headlines and titles are evaluated using the ROUGE score by comparing them to the reference output.

**Application to LiveCodeBench.** The LiveCodeBench experiment uses 612 programming problems. We follow the official LiveCodeBench code generation prompt and evaluation format. For each problem, we configure a chat-style prompt that contains the problem statement. Once the starter code is provided, it will be included in the prompt. Depending on the official prompt format, we use different generative demonstrations for the function completion problem and the standard input problem. Each model is instructed to generate the Python code contained in the code block. We set the temperature at 1.0 and sampled  $N = 5$

<sup>6</sup><https://github.com/EleutherAI/lm-evaluation-harness>

<sup>7</sup><https://huggingface.co/datasets/meta-LLaMA/LLaMA-3.1-8B-Instruct-evals>

Table 7: **Paired significance tests against the strongest baselines.** Two-sided paired  $t$ -tests on the seven-task average accuracy over the three matched runs of Table 1.

| Generator    | Comparison         | $\Delta$ | 95% CI        | $p$   |
|--------------|--------------------|----------|---------------|-------|
| GPT-4o       | vs. CoT-WP         | +1.89    | [0.87, 2.90]  | .0153 |
|              | vs. Self-Certainty | +1.86    | [0.66, 3.06]  | .0218 |
| GPT-4o-mini  | vs. CoT-WP         | +2.08    | [1.22, 2.94]  | .0091 |
|              | vs. Self-Certainty | +1.71    | [0.36, 3.07]  | .0322 |
| LLaMA-3.1-8B | vs. CoT-WP         | +1.90    | [0.33, 3.47]  | .0349 |
|              | vs. Self-Certainty | +1.23    | [-0.06, 2.53] | .0547 |

Table 8: **Paired significance tests for the scoring components.** Two-sided paired  $t$ -tests of ReFeri against each single-term variant, on the matched aggregate over the three runs of Table 25.

| Comparison           | 95% CI         | $p$   |
|----------------------|----------------|-------|
| ReFeri vs. Forward   | [0.021, 0.155] | .0302 |
| ReFeri vs. Backward  | [3.250, 5.859] | .0044 |
| ReFeri vs. -Backward | [3.796, 6.826] | .0044 |

candidate programs per problem. The selected response is considered correct only when it passes both public and hidden test cases. Thus, all selection methods operate on the same set of candidate programs, and test-case execution is used only for final evaluation.

## B More Quantitative Results

In this section, we use the candidate responses generated in the first run of Table 1 unless otherwise specified.

### B.1 Significance tests over matched runs

To evaluate whether the observed performance improvements are robust to variability across runs, we perform two-sided paired  $t$ -tests on three matched generative runs. For each run, methods are evaluated on the same candidate pool, and we compare their seven-task average accuracy. Table 7 shows the comparison results with CoT-WP and Self-Certainty for all three generators. ReFeri outperforms the others in all six comparisons; five of these are statistically significant at the  $p < .05$  level, while the comparison with Self-Certainty on LLaMA-3.1-8B yields a  $p=.0547$  result. We apply the same procedure to the scoring-related ablation experiments reported in Table 25. As reported in Table 8, the overall ReFeri score outperforms both the forward-only variant ( $p=.0302$ ) and the two backward-only variants ( $p=.0044$ ).

### B.2 Additional comparison with few-shot prompting-based selection methods

Among the multiple answer selection methods, the simplest and most accessible approach (*e.g.*, learning overhead, domain specificity, etc.) is arguably LLM-as-Judge (Chen et al., 2023; Zheng et al., 2023). It uses the LLM itself to score and select answers via in-context learning without any additional training or external verifiers. In particular, the addition of few-shot examples to LLM-as-Judge might be most closely aligned with the core motivation of ReFeri, which is to use demonstrations not only for generation but also for validation. Therefore, in this section, we compare ReFeri and (1) the original USC (Chen et al., 2023), (2) *USC with few-shot* (our adaptation), and (3) *LLM-as-Judge with few-shot* created with our optimized prompt (see list 8 and 9). For a fair comparison and to maintain consistency with Table 1, LLaMA-3.1-8B-Instruct is employed as the judging model for all LLM-as-Judge variants.

As shown in Table 9, ReFeri consistently achieves the best or second-best accuracy across all LLMs and benchmarks. Interestingly, we observe that adding few-shot demonstrations to USC often degrades performance (*e.g.*, on GPQA and DROP with GPT-4o-mini and LLaMA-3.1-8B), which is likely due to the sensitivity of LLMs to prompt format and positional bias of the responses.

Notably, we observe that both prompt-based selection methods, USC and LLM-as-Judge, are highly sensitive to the order of candidate responses. In our experiments, USC frequently selected one of the first two responses regardless of correctness; on multiple choice question tasks this pattern is less extreme but skew toward early positions is still visible. Moreover, since USC requires explicit answer extraction, tasks such as MuSR-ta revealed many failure cases (*e.g.*, over 40% failures in Table 10), further highlighting its fragility. This highlights a critical weakness in prompt-based selection: the output is often determined more by position than content. Based on these observations, we conducted additional experiments where we randomly rearranged the order of candidate responses. Indeed, we observed this issue in Table 11; on GPQA, for example, USC’s accuracy varied notably across different permutations (*e.g.*, 46.0  $\rightarrow$  41.9), demonstrating its sensitivity to presentation order. In contrast, our approach mitigates such ordering artifacts by decoupling few-shot demon-

Table 9: **Comparison with prompting-based selection.** Overall performance on seven reasoning benchmarks comparing the proposed ReFeri with different prompting-based baselines not require additional training, under three different state-of-the-art LLMs. For reference, we additionally include the Oracle upper bound.

| Models       | Methods       | MuSR-ta     | MuSR-op     | GPQA        | MATH500     | DROP                      | HotpotQA                  | MMLU-PRO    | Avg.        |
|--------------|---------------|-------------|-------------|-------------|-------------|---------------------------|---------------------------|-------------|-------------|
| GPT-4o       | USC           | 85.2        | <u>71.1</u> | <u>47.0</u> | <u>77.4</u> | 82.2 / 90.2               | <u>45.6</u> / 59.7        | <u>74.5</u> | <u>69.0</u> |
|              | USC-w/Fewshot | 88.8        | 69.1        | 46.0        | <u>77.4</u> | 82.0 / 89.9               | 45.4 / 60.1               | 74.1        | <u>69.0</u> |
|              | LLM-as-Judge  | 86.0        | 68.0        | 46.5        | <b>77.8</b> | 82.8 / 91.0               | 45.6 / 59.8               | 73.3        | 68.6        |
|              | ReFeri        | <b>90.4</b> | <b>71.9</b> | <b>51.5</b> | <b>77.8</b> | <b>83.6 / 91.1</b>        | <b>47.0 / 60.7</b>        | <b>75.4</b> | <b>71.1</b> |
|              | Oracle        | 96.4        | 87.9        | 72.2        | 86.6        | 89.2 / 94.7               | 55.4 / 69.2               | 84.1        | 81.7        |
| GPT-4o-mini  | USC           | 74.4        | 60.9        | <b>46.0</b> | 77.8        | 76.8 / 83.8               | <u>35.0</u> / <u>47.2</u> | <u>63.7</u> | <u>62.1</u> |
|              | USC-w/Fewshot | <u>76.4</u> | <b>63.3</b> | 39.9        | <b>78.2</b> | 77.2 / 84.0               | 34.8 / 46.6               | 63.2        | 61.9        |
|              | LLM-as-Judge  | 75.6        | 60.6        | 34.3        | 77.0        | <u>77.4</u> / <u>84.4</u> | <u>35.0</u> / 46.7        | 63.3        | 60.5        |
|              | ReFeri        | <b>82.8</b> | <u>61.3</u> | <u>41.9</u> | <u>77.8</u> | <b>79.2 / 84.9</b>        | <b>36.2 / 48.0</b>        | <b>64.9</b> | <b>63.4</b> |
|              | Oracle        | 97.2        | 78.1        | 70.7        | 85.8        | 86.6 / 91.6               | 43.4 / 56.4               | 76.8        | 76.9        |
| LLaMA-3.1-8B | USC           | 67.2        | 52.3        | <u>28.8</u> | <u>49.6</u> | <b>69.6 / 75.8</b>        | 24.4 / 32.5               | <b>45.6</b> | 48.2        |
|              | USC-w/Fewshot | <u>70.0</u> | 53.9        | 28.3        | 47.8        | 69.0 / 75.3               | <b>25.2</b> / <u>32.3</u> | <u>45.1</u> | 48.5        |
|              | LLM-as-Judge  | 65.2        | 55.1        | 21.2        | 46.0        | 67.7 / 74.0               | 23.4 / 31.2               | 44.1        | 46.1        |
|              | ReFeri        | <b>79.6</b> | <b>57.8</b> | <b>35.4</b> | <b>51.2</b> | <u>69.4</u> / <u>75.7</u> | <u>25.0</u> / <b>33.2</b> | <u>45.1</u> | <b>51.9</b> |
|              | Oracle        | 97.6        | 88.3        | 59.6        | 66.6        | 83.4 / 88.8               | 33.8 / 45.0               | 70.8        | 71.4        |

strations from the selection prompt and using them only for scoring. Furthermore, LLM-as-Judge does not perform reliably on more complex tasks (*e.g.*, GPQA showing a noticeable accuracy degradation compared to other methods). These results emphasize that naively incorporating a few examples into prompts does not guarantee consistent gains, and that ReFeri is more robust and scalable. Finally, we note that the application of prompt-based approach could be limited due to inherent input context-window length. For reference, we also report an oracle upper bound in Table 9. This represents the accuracy achieved when selecting the optimal response from  $N$  samples. This serves purely as a ceiling to describe in context how close each method approaches the maximum achievable performance. Although the gap with this ceiling is still noticeable, this highlights meaningful room for future improvements.

### B.3 Complementarity with majority voting

As denoted in Section 1, *Self-Consistency* (majority voting) has an inherent limitation: it is only applicable when answers can be easily extracted and normalized for voting. For example, it is hard to be applied for open-ended text generation. For this reason, rather than the original self-consistency (Wang et al., 2023b), we mainly consider Universal Self-Consistency (USC) (Chen et al., 2023) as a baseline, which uses prompt-based evaluation to select among free-form outputs. In fact, HotpotQA is a representative case where standard self-consistency cannot be reliably applied, since mapping free-form outputs into consistent discrete

answer categories is non-trivial.

Although standard Self-Consistency is difficult to apply in open-ended tasks, we conducted additional evaluations using standard self-consistency. To enable it on HotpotQA, we identified overlapping shared spans among free-form responses, counted how many responses contain each span, and used these aggregated counts as voting scores. In addition, inspired by the weighted voting scheme in Self-Certainty (Kang et al., 2025), we experimented with a Borda-voting method that uses the ReFeri metric as the weight, fixing the parameter  $p = 1$  to avoid introducing new hyperparameters. For notational simplicity, in this subsection we let  $r_n$  denote the normalized extracted answer from the  $n$ -th candidate response. Namely, the standard Self-Consistency selects the answer with the highest vote,

$$n_{sc}^* = \arg \max_{n \in \{1, \dots, N\}} \sum_{j=1}^N \mathbf{1}[r_j = r_n],$$

in which each candidate contributes the same weight to one. The Borda vote generalizes this formula by replacing uniform unit weights as follows with rank-based weights derived from the ReFeri score. Let  $\text{rank}(n)$  denote the rank of candidate  $r_n$  based on its ReFeri score. The corresponding borda weight is calculated as follows:

$$w_n = (N - \text{rank}(n) + 1)^p$$

The final selection score of candidate  $n$  is obtained by summing the weights of all candidates who

Table 10: **Response selection distribution per task (GPT-4o-mini).** The selections distribution suggests that prompt-based selection method may exhibit position bias rather than relying solely on response quality.

| Task     | Method         | #1   | #2   | #3   | #4   | #5   | Fail (-1) |
|----------|----------------|------|------|------|------|------|-----------|
| MATH500  | USC            | 90.2 | 3.8  | 1.0  | 2.8  | 2.2  | 0.0       |
|          | USC-w/ Fewshot | 81.0 | 12.6 | 0.6  | 3.2  | 2.6  | 0.0       |
|          | LLM-as-Judge   | 51.4 | 2.2  | 8.0  | 12.2 | 25.4 | 0.8       |
| MMLU-Pro | USC            | 34.2 | 19.1 | 7.2  | 24.0 | 15.5 | 0.0       |
|          | USC-w/ Fewshot | 18.8 | 31.0 | 7.8  | 22.7 | 19.7 | 0.0       |
|          | LLM-as-Judge   | 22.1 | 8.0  | 7.8  | 13.5 | 48.3 | 1.2       |
| GPQA     | USC            | 21.7 | 15.2 | 13.1 | 23.7 | 26.3 | 0.0       |
|          | USC-w/ Fewshot | 19.7 | 17.7 | 10.6 | 23.7 | 28.3 | 0.0       |
|          | LLM-as-Judge   | 30.8 | 9.1  | 10.1 | 7.6  | 42.4 | 0.0       |
| DROP     | USC            | 73.8 | 21.8 | 1.8  | 1.2  | 1.4  | 0.0       |
|          | USC-w/ Fewshot | 78.2 | 16.6 | 3.0  | 1.0  | 1.2  | 0.0       |
|          | LLM-as-Judge   | 68.8 | 9.6  | 4.6  | 4.4  | 12.4 | 0.2       |
| HotpotQA | USC            | 77.0 | 13.4 | 2.8  | 5.4  | 1.4  | 0.0       |
|          | USC-w/ Fewshot | 68.6 | 20.6 | 2.8  | 6.0  | 2.0  | 0.0       |
|          | LLM-as-Judge   | 65.0 | 15.0 | 4.8  | 6.0  | 9.2  | 0.0       |
| MuSR-op  | USC            | 51.6 | 18.0 | 11.7 | 7.4  | 11.3 | 0.0       |
|          | USC-w/ Fewshot | 36.7 | 40.2 | 10.2 | 6.2  | 6.6  | 0.0       |
|          | LLM-as-Judge   | 26.9 | 21.5 | 15.2 | 20.7 | 15.2 | 0.4       |
| MuSR-ta  | USC            | 34.0 | 3.2  | 0.8  | 9.6  | 9.6  | 42.8      |
|          | USC-w/ Fewshot | 46.4 | 35.2 | 2.4  | 3.6  | 12.4 | 0.0       |
|          | LLM-as-Judge   | 27.6 | 2.8  | 0.4  | 1.2  | 50.4 | 17.6      |

Table 11: **Evaluation of USC ordering with GPT-4o-mini.** We evaluate USC under the default candidate order and two random permutations (*perm-A*, *perm-B*). The variation across permutations shows that prompt-based judgment can be sensitive to presentation order.

| Methods       | MuSR-ta | MuSR-op | GPQA | MATH500 | DROP        | HotpotQA    | MMLU-PRO | Avg. |
|---------------|---------|---------|------|---------|-------------|-------------|----------|------|
| USC (perm-A)  | 75.6    | 59.8    | 41.9 | 78.0    | 76.6 / 83.5 | 35.6 / 47.2 | 63.2     | 61.5 |
| USC (perm-B)  | 77.2    | 56.6    | 45.0 | 77.2    | 76.6 / 83.4 | 35.2 / 46.9 | 63.6     | 61.6 |
| USC (default) | 74.4    | 60.9    | 46.0 | 77.8    | 76.8 / 83.8 | 35.0 / 47.2 | 63.7     | 62.1 |

make the same prediction:

$$k_{\text{Borda}}^* = \arg \max_{n \in \{1, \dots, N\}} \sum_{j=1}^N w_j * \mathbf{1}[r_j = r_n],$$

The results are presented in the Table 12. Overall, ReFeri continues to outperform self-consistency on average. For instance, on MuSR-ta with LLaMA, ReFeri shows substantially higher performance (*e.g.*, 79.6 vs. 76.4). More importantly, these methods are complementary: applying Borda voting with ReFeri yields notable improvements over conventional self-consistency, particularly on LLaMA.

#### B.4 Applying ReFeri to zero-shot responses

As shown in Table 1, Zero-shot CoT often achieves higher accuracy than Few-shot CoT, reflecting the intrinsic knowledge of the model. Although we primarily apply ReFeri to Few-shot CoT candidates,

the same scoring function in Eq. 11 can also be applied to reasoning paths generated by Zero-shot CoT. We focus on Few-shot CoT since it usually yields better reasoning paths, as shown in Table 1. With the experiments in Table 13, we verify that applying ReFeri to Zero-shot CoT yields improvements. These results further suggest that the few-shot exemplars in ReFeri mainly function as a post-hoc validation pipeline, rather than as generation guidance as in conventional Few-shot CoT. Also, this effectiveness of ReFeri under decoupling between generation and selection suggests a robust alternative to conventional few-shot prompting strategies, particularly in settings where few-shot examples are ineffective with LLMs.

Table 12: **Complementarity with self-consistency.** We compare standard Self-Consistency with ReFeri and a rank-weighted voting variant that incorporates ReFeri selection scores. The results show that ReFeri is competitive and can further complement majority voting.

| Models       | Methods           | MuSR-ta     | MuSR-op     | GPQA        | MATH500     | DROP               | HotpotQA           | MMLU-PRO    | Avg.        |
|--------------|-------------------|-------------|-------------|-------------|-------------|--------------------|--------------------|-------------|-------------|
| LLaMA-3.1-8B | Self-Consistency  | 76.4        | <u>54.7</u> | 26.3        | <u>52.0</u> | <u>73.0 / 78.0</u> | 23.0 / 29.8        | <u>45.1</u> | 50.1        |
|              | ReFeri            | <b>79.6</b> | <b>57.8</b> | <b>35.4</b> | 51.2        | 69.4 / 75.7        | <u>25.0 / 33.2</u> | <u>45.1</u> | <u>51.9</u> |
|              | + Borda-vote(p=1) | <u>78.4</u> | <b>57.8</b> | <u>30.3</u> | <b>53.2</b> | <b>74.0 / 79.7</b> | <b>26.8 / 34.9</b> | <b>46.4</b> | <b>52.4</b> |
| GPT-4o-mini  | Self-Consistency  | <u>84.8</u> | 60.6        | <u>43.4</u> | <b>79.6</b> | <u>80.0 / 85.3</u> | 35.6 / 46.8        | <u>65.3</u> | <u>64.2</u> |
|              | ReFeri            | 82.8        | <b>61.3</b> | 41.9        | 77.8        | 79.2 / 84.9        | <u>36.2 / 48.0</u> | 64.9        | 63.4        |
|              | + Borda-vote(p=1) | <b>86.8</b> | <u>60.9</u> | <b>44.4</b> | <u>79.4</u> | <b>80.4 / 86.0</b> | <b>36.8 / 48.3</b> | <b>65.7</b> | <b>64.9</b> |
| GPT-4o       | Self-Consistency  | 86.8        | <u>71.5</u> | <u>50.5</u> | 80.0        | <u>84.4 / 91.6</u> | 45.6 / 60.5        | <b>76.4</b> | 70.7        |
|              | ReFeri            | <b>90.4</b> | <b>71.9</b> | <b>51.5</b> | 77.8        | 83.6 / 91.1        | <u>47.0 / 60.7</u> | 75.4        | <u>71.1</u> |
|              | + Borda-vote(p=1) | <u>88.8</u> | <u>71.5</u> | 49.5        | <b>80.2</b> | <b>84.8 / 92.0</b> | <b>48.0 / 61.9</b> | <u>76.2</u> | <b>71.3</b> |

Table 13: **Applying ReFeri to zero-shot responses.** We apply the same ReFeri scoring function to candidates generated by Zero-shot CoT. These results show that ReFeri can also improve response selection when demonstrations are used only for verification.

| Models       | Methods   | MATH500     | GPQA        | MuSR-ta     |
|--------------|-----------|-------------|-------------|-------------|
| GPT-4o-mini  | Zero-shot | 76.4        | 43.0        | 56.2        |
|              | ReFeri    | <b>78.2</b> | <b>43.9</b> | <b>58.8</b> |
| GPT-4o       | Zero-shot | 77.5        | 48.8        | 66.6        |
|              | ReFeri    | <b>80.8</b> | <b>54.0</b> | <b>69.6</b> |
| LLaMA-3.1-8B | Zero-shot | 44.2        | 21.6        | 39.6        |
|              | ReFeri    | <b>50.8</b> | <b>24.2</b> | <b>41.2</b> |

Table 14: **Results of different few-shot example sets.** We evaluate GPT-4o-mini under three different few-shot example sets. While absolute performance varies with the quality of the demonstrations and the resulting candidate pool, ReFeri consistently improves over the corresponding Few-shot CoT.

| Methods    | MATH500     | GPQA        | MuSR-ta     |
|------------|-------------|-------------|-------------|
| Few-shot 1 | 75.2        | 41.3        | 77.0        |
| ReFeri 1   | <b>77.8</b> | <b>41.9</b> | <b>82.8</b> |
| Few-shot 2 | 74.5        | 41.5        | 57.8        |
| ReFeri 2   | <b>79.0</b> | <b>43.4</b> | <b>59.2</b> |
| Few-shot 3 | 75.0        | 38.9        | 60.1        |
| ReFeri 3   | <b>77.8</b> | <b>41.9</b> | <b>62.8</b> |

## B.5 Robustness to few-shot examples

ReFeri highly relies on few-shot examples for scoring of both forward and backward terms (Section 2.2). This raises the question of how sensitive the method is to the choice of few-shot examples. To answer this, we conduct a sensitivity study on MATH500, GPQA and MuSR-ta using GPT-4o-mini, where we use three different few-shot examples with one original and two newly sampled. As shown in Table 14, both Few-shot CoT and ReFeri show variation across these different sets. Interest-

Table 15: **Performance under weak few-shot examples.** We evaluate ReFeri when the few-shot examples are intentionally weaker than the original. Although weaker demonstrations reduce the quality of the generated candidate pool, ReFeri improves over the corresponding Few-shot CoT across generation models.

| Models       | Methods  | MATH500     | GPQA        | MuSR-ta     |
|--------------|----------|-------------|-------------|-------------|
| GPT-4o-mini  | Few-shot | 73.7        | 41.1        | 57.5        |
|              | ReFeri   | <b>76.4</b> | <b>43.4</b> | <b>58.0</b> |
| GPT-4o       | Few-shot | 75.2        | 46.1        | 71.0        |
|              | ReFeri   | <b>79.0</b> | <b>47.0</b> | <b>75.6</b> |
| LLaMA-3.1-8B | Few-shot | 39.5        | 26.6        | 38.7        |
|              | ReFeri   | <b>47.6</b> | <b>33.3</b> | <b>41.6</b> |

Table 16: **Quality judgment scores of weak few-shot examples.** GPT-4o judges the original and weak few-shot examples on a 1–10 scale.

| Judge by GPT-4o (1–10) | MATH500 | GPQA | MuSR-ta |
|------------------------|---------|------|---------|
| Few-shot               | 8       | 8    | 8       |
| Low quality            | 3       | 5    | 4       |

ingly, MuSR-ta once again highlights the importance of example quality; when synthesizing new data according to (Sprague et al., 2024) to use as few-shot examples, baseline accuracy significantly degrades. Nevertheless, ReFeri demonstrates consistent performance improvements. These results indicate that ReFeri remains robust to exemplar choice and is consistently effective, rather than overfitted to specific demonstrations.

We believe that constructing accurate few-shot examples is a minimal effort that one should invest to guide LLMs (even humans) toward a proper behavior for a target task. However, in practical applications, the clarity of few-shot example may not always be guaranteed. To evaluate robustness, we first synthesize low-quality few-shot examples by converting the original examples via prompting

GPT-4o-mini to degrade the quality of reasoning in data. As shown in Table 15, ReFeri maintains consistent improvements even under degraded exemplars, indicating that recycled few-shot examples during verification remains effective without well curated examples; for instance, on MATH500 with LLaMA-3.1-8B, accuracy improves from 39.5% to 47.6% (+8.1), demonstrating ReFeri’s robustness. Overall, these results confirm that ReFeri remains effective even with suboptimal examples and can provide meaningful gains in more practical, less curated scenarios.

To verify the degradation, we asked GPT-4o to evaluate the quality of the original set and the weak example set. The evaluation was conducted in random order, and information about each set was not provided to avoid bias. As shown in Table 16, the weak set consistently received significantly lower scores (3–5 points) compared to the original examples (8 points). This further demonstrates that ReFeri maintains its effectiveness even when the quality of the provided examples is low. See list 10-11 for exact prompts used in the generation and evaluation assessment.

### B.6 Robustness to generation and evaluation prompts

Additionally, to investigate the impact of various prompt choices, we conduct new experiments with two alternative prompting techniques, following prior work planning and role-playing (Wang et al., 2023a; Kong et al., 2024). Specifically, we assess the robustness of ReFeri by varying prompts during both the generation stage and the verification stage by adapting different prompting styles (orig, plan, and role). For plan and role prompting at the generation, we newly sample five responses similar to Table 1. Table 17 shows the performance on MATH500, GPQA and MuSR-ta under different configurations. A key observation is that verification performance remains highly consistent across different verification prompt styles for a given generation prompt style. For instance, on GPQA, performance for "plan  $\rightarrow$  orig" and "plan  $\rightarrow$  plan" conditions is identical (47.5 vs. 47.5), with similar consistency observed for the "role" condition (47.5 vs. 47.0). This indicates that ReFeri is inherently robust to variations in prompt formatting during the evaluation stage.

However, we also observe that the initial quality of the generated candidate set varies depending on the prompt style. For relatively simple tasks like

MATH500, the quality of generated responses is similar across prompts. Conversely, on the more challenging GPQA, prompts offering structured guidance (*e.g.*, plans or roles) tend to generate higher-quality seeds, reflected in slightly higher accuracy. Consequently, ReFeri performs better when the initial candidates are of higher quality.

### B.7 Robustness to estimation model calibration

In the main experiments, likelihood scores are computed from the estimator’s original logits, corresponding to  $T = 1.0$ . A natural question is how sensitive ReFeri is to the calibration of the estimator. To examine sensitivity to estimator calibration, we apply post-hoc temperature scaling to the logits before computing token-level likelihoods, using  $p_T(\cdot | x) = \text{softmax}(z/T)$  with  $T \in \{0.5, 1.0 \text{ (default)}, 1.5, 2.0\}$ . As shown in Table 18, ReFeri remains remarkably robust across temperatures, showing only minimal variation even under substantial overconfidence or underconfidence.

This robustness is further demonstrated through cross-model and cross-scale evaluations. As shown in Figure 5 and Appendix B.9 (Table 21), ReFeri using the 1B estimation model maintains stable performance across various tasks and calibration environments, whereas the likelihood-based baseline CoT-WP exhibits significantly greater variability. These observations suggest that ReFeri does not heavily rely on the precise calibration of the estimation model. The backward term provides an additional signal that helps compensate for calibration drift by capturing explanatory alignment with the few-shot examples rather than relying solely on raw likelihood peaks.

### B.8 Robustness to nearest neighbor approximation

To further validate our nearest-neighbor approximation, we conduct two additional experiments. First, we evaluate the robustness of ReFeri to fixed single-exemplar selection. For this purpose, we analyze two settings: a worst-case setting that uses the lowest-performing exemplar, and an oracle upper-bound setting in which an output is considered correct if any single exemplar produces the correct answer. As shown in Table 19, even the worst-case fixed-exemplar setting achieves competitive performance, while our proposed semantic nearest-neighbor selection outperforms the worst-case set-

Table 17: **Robustness to generation and evaluation prompts.** We vary the prompt style used to generate candidate responses and the prompt style used for verification. Differences across generation prompts mainly reflect changes in the quality of the initial candidate pool.

| Gen Prompt | Eval Prompt | MATH500  |             | GPQA     |             | MuSR-ta  |             |
|------------|-------------|----------|-------------|----------|-------------|----------|-------------|
|            |             | Few-shot | ReFeri      | Few-shot | ReFeri      | Few-shot | ReFeri      |
| Orig       | Orig        | 75.2     | <b>77.8</b> | 41.3     | <b>41.9</b> | 77.0     | <b>82.8</b> |
|            | Plan        | 75.2     | <b>78.0</b> | 41.3     | <b>42.4</b> | 77.0     | <b>82.8</b> |
|            | Role        | 75.2     | <b>77.8</b> | 41.3     | <b>41.9</b> | 77.0     | <b>82.4</b> |
| Plan       | Plan        | 74.6     | <b>78.2</b> | 42.6     | <b>47.5</b> | 77.0     | <b>82.4</b> |
|            | Orig        | 74.6     | <b>78.4</b> | 42.6     | <b>47.5</b> | 77.0     | <b>82.4</b> |
| Role       | Role        | 74.5     | <b>78.2</b> | 43.5     | <b>47.5</b> | 75.8     | <b>81.6</b> |
|            | Orig        | 74.5     | <b>78.2</b> | 43.5     | <b>47.0</b> | 75.8     | <b>81.6</b> |

Table 18: **Robustness to estimation model calibration.** We report the performance of ReFeri across different temperature scaling factors  $T \in \{0.5, 1.0, 1.5, 2.0\}$  applied to the estimation model before computing token-level likelihoods. The results show that ReFeri is largely stable across calibration settings.

| Models       | Temp | MuSR-ta     | MuSR-op     | GPQA        | MATH500     | DROP               | HotpotQA           | Avg.        |
|--------------|------|-------------|-------------|-------------|-------------|--------------------|--------------------|-------------|
| GPT-4o       | 0.5  | <u>90.4</u> | <u>71.1</u> | <b>52.0</b> | 77.4        | <b>84.0 / 91.3</b> | <b>47.6 / 60.7</b> | <b>70.4</b> |
|              | 1    | <u>90.4</u> | <b>71.9</b> | <u>51.5</u> | <b>77.8</b> | <u>83.6 / 91.1</u> | <u>47.0 / 60.7</u> | <b>70.4</b> |
|              | 1.5  | <b>91.2</b> | <u>71.1</u> | 50.5        | <u>77.6</u> | <u>83.4 / 91.1</u> | <u>45.0 / 59.2</u> | <u>69.8</u> |
|              | 2    | <u>90.4</u> | <b>71.9</b> | 51.0        | <u>77.6</u> | 81.8 / 89.9        | <u>44.8 / 59.3</u> | 69.6        |
| GPT-4o-mini  | 0.5  | 82.0        | 59.4        | <b>43.9</b> | <u>78.0</u> | <u>78.0 / 83.8</u> | <u>35.8 / 47.1</u> | 62.9        |
|              | 1    | 82.8        | <b>61.3</b> | 41.9        | <u>77.8</u> | <b>79.2 / 84.9</b> | <u>36.2 / 48.0</u> | 63.2        |
|              | 1.5  | <b>83.6</b> | <u>60.2</u> | <u>43.4</u> | <u>78.0</u> | <u>78.0 / 84.7</u> | <b>36.4 / 48.3</b> | <b>63.3</b> |
|              | 2    | <u>83.2</u> | 59.0        | 41.4        | <b>78.4</b> | <u>77.4 / 84.1</u> | <u>35.2 / 47.6</u> | 62.4        |
| LLaMA-3.1-8B | 0.5  | <b>79.6</b> | <u>57.0</u> | 33.3        | 50.6        | <u>69.8 / 75.9</u> | <u>24.6 / 33.0</u> | 52.5        |
|              | 1    | <b>79.6</b> | <b>57.8</b> | <b>35.4</b> | 51.2        | <u>69.4 / 75.7</u> | <u>25.0 / 33.2</u> | <b>53.1</b> |
|              | 1.5  | <u>78.8</u> | <u>57.0</u> | <u>34.8</u> | <b>52.0</b> | <u>70.2 / 76.1</u> | <b>25.2 / 33.2</b> | <u>53.0</u> |
|              | 2    | 78.0        | <u>56.2</u> | 34.3        | <u>51.8</u> | <b>71.0 / 77.1</b> | <u>24.6 / 32.7</u> | <u>52.7</u> |

ting in task average for every model. Second, we also conducted experiments using GPT-4o-mini to measure whether utilizing multiple nearest neighbors (top-k) enhances performance. We compared the top-1 nearest neighbor against top-2 and top-3 neighbors. As shown in Table 20, the average scores across benchmarks remained highly stable (63.4, 63.3, and 63.5, respectively). This confirms that utilizing multiple examples yields marginal differences, validating the efficiency and effectiveness of our top-1 design choice. Overall, these results demonstrate that nearest-neighbor approximation is an efficient and effective design choice for ReFeri.

## B.9 Robustness to different estimation models and computational efficiency

Table 21 provides a full detailed breakdown of ReFeri across all generation–estimation model combinations of MATH500, GPQA and MuSR-ta. This complements the average performance across different generation LLMs (GPT-4o-mini, GPT-4o,

and LLaMA3.1-8B) shown in Figure 5. Across all three tasks, ReFeri shows consistent performance gains regardless of the estimation model used, emphasizing its robustness. At the same time, we observe task-dependent estimator trends: Qwen-2.5-7B performs best on MATH500, whereas LLaMA-3.1-70B performs best on GPQA and MuSR-ta, as discussed in Section 3.3. Notably, even the smaller model (LLaMA-3.2-1B) performs competitively on (relatively) simple tasks such as MATH500, on average, indicating that large estimator capacity is not always necessary for relatively familiar or structured reasoning tasks. These results suggest that different estimators may align better with different reasoning styles, while the proposed few-shot-based verification strategy remains broadly stable.

While the lightweight approximation offers significant computational advantages, replacing the entire Bayesian term in Eq. 5 with the single most relevant example implies a theoretical simplifi-

Table 19: **Robustness analysis of fixed exemplar selection.** We compare ReFeri under three exemplar-selection settings: the lowest-performing fixed exemplar (worst), the oracle upper bound over fixed exemplars (best), and our semantic nearest-neighbor selection strategy (ours).

| Models       | Methods              | MuSR-ta<br>(Acc.) | MuSR-op<br>(Acc.) | GPQA<br>(Acc.) | MATH500<br>(Acc.) | DROP<br>(EM / F1)  | HotpotQA<br>(EM / F1) | MMLU-PRO<br>(Acc.) | Avg.        |
|--------------|----------------------|-------------------|-------------------|----------------|-------------------|--------------------|-----------------------|--------------------|-------------|
| GPT-4o       | ReFeri (worst)       | 90.0              | 70.7              | 51.0           | 77.8              | 83.4 / 91.2        | 46.4 / 60.2           | 75.3               | 70.7        |
|              | ReFeri (best)        | <b>90.8</b>       | <b>73.4</b>       | <b>51.5</b>    | <b>78.8</b>       | <b>84.2 / 91.5</b> | <b>47.8 / 61.7</b>    | <b>76.1</b>        | <b>71.8</b> |
|              | <b>ReFeri (ours)</b> | <u>90.4</u>       | <u>71.9</u>       | <u>51.5</u>    | <u>77.8</u>       | <u>83.6 / 91.1</u> | <u>47.0 / 60.7</u>    | <u>75.4</u>        | <u>71.1</u> |
| GPT-4o-mini  | ReFeri (worst)       | <u>82.8</u>       | 60.2              | 41.4           | <u>77.8</u>       | 79.0 / 84.7        | 35.4 / 47.5           | <u>64.9</u>        | 63.1        |
|              | ReFeri (best)        | <b>83.2</b>       | <b>61.7</b>       | <b>42.4</b>    | <b>78.2</b>       | <b>79.8 / 85.5</b> | <b>36.6 / 48.8</b>    | <b>65.8</b>        | <b>64.0</b> |
|              | <b>ReFeri (ours)</b> | <u>82.8</u>       | <u>61.3</u>       | <u>41.9</u>    | <u>77.8</u>       | <u>79.2 / 84.9</u> | <u>36.2 / 48.0</u>    | <u>64.9</u>        | <u>63.4</u> |
| Llama-3.1-8B | ReFeri (worst)       | 78.8              | 55.9              | 33.8           | 50.8              | 69.8 / 76.0        | 24.6 / 32.8           | 44.8               | 51.2        |
|              | ReFeri (best)        | <b>81.6</b>       | <b>60.2</b>       | <b>35.4</b>    | <b>51.8</b>       | <b>71.2 / 77.1</b> | <b>25.6 / 33.9</b>    | <b>46.5</b>        | <b>53.2</b> |
|              | <b>ReFeri (ours)</b> | <u>79.6</u>       | <u>57.8</u>       | <u>35.4</u>    | <u>51.2</u>       | <u>69.4 / 75.7</u> | <u>25.0 / 33.2</u>    | <u>45.1</u>        | <u>51.9</u> |

Table 20: **Robustness analysis of increasing the number of nearest neighbors.** We evaluate ReFeri using the top- $k$  nearest exemplars across three generation models. This confirms that utilizing multiple examples yields marginal differences, validating the efficiency and effectiveness of our top-1 design choice.

| Model        | Methods | MuSR-ta<br>(Acc.) | MuSR-op<br>(Acc.) | GPQA<br>(Acc.) | MATH500<br>(Acc.) | DROP<br>(EM / F1)  | HotpotQA<br>(EM / F1) | MMLU-PRO<br>(Acc.) | Avg.        |
|--------------|---------|-------------------|-------------------|----------------|-------------------|--------------------|-----------------------|--------------------|-------------|
| GPT-4o       | Top-1   | <u>90.4</u>       | <b>71.9</b>       | <b>51.5</b>    | 77.8              | <b>83.6 / 91.1</b> | <u>47.0 / 60.7</u>    | <b>75.4</b>        | <u>71.1</u> |
|              | Top-2   | <b>90.8</b>       | <b>71.9</b>       | <b>51.5</b>    | <u>78.0</u>       | <b>83.6 / 91.1</b> | <u>47.0 / 60.7</u>    | <b>75.4</b>        | <b>71.2</b> |
|              | Top-3   | <b>90.8</b>       | <u>70.7</u>       | <u>51.0</u>    | <b>78.2</b>       | <b>83.6 / 91.2</b> | <b>47.2 / 60.8</b>    | <b>75.4</b>        | 71.0        |
| GPT-4o-mini  | Top-1   | <b>82.8</b>       | <b>61.3</b>       | <u>41.9</u>    | <b>77.8</b>       | 79.2 / 84.9        | <b>36.2 / 48.0</b>    | <u>64.9</u>        | <u>63.4</u> |
|              | Top-2   | <b>82.8</b>       | 60.9              | <u>41.9</u>    | <b>77.8</b>       | <u>79.4 / 85.1</u> | 35.6 / 47.4           | <u>64.9</u>        | 63.3        |
|              | Top-3   | <b>82.8</b>       | <b>61.3</b>       | <b>42.4</b>    | <b>77.8</b>       | <b>79.6 / 85.3</b> | <u>35.8 / 47.7</u>    | <b>65.0</b>        | <b>63.5</b> |
| LLaMA-3.1-8B | Top-1   | <b>79.6</b>       | <b>57.8</b>       | <b>35.4</b>    | 51.0              | 69.4 / 75.7        | <b>25.0 / 33.2</b>    | <b>45.1</b>        | <b>51.9</b> |
|              | Top-2   | <b>79.6</b>       | <u>56.2</u>       | 34.3           | <b>51.2</b>       | <u>69.6 / 76.0</u> | <u>24.6 / 32.8</u>    | <u>45.0</u>        | 51.5        |
|              | Top-3   | <b>79.6</b>       | 55.9              | <u>34.8</u>    | 50.8              | <b>70.2 / 76.4</b> | <u>24.6 / 32.8</u>    | <u>45.0</u>        | <u>51.6</u> |

Table 21: **Robustness to different estimation models.** Full results with different estimation models on MATH500, GPQA, and MuSR-ta. ReFeri remains effective across estimator sizes and backbones.

| (a) MATH500   |             |        |              |      |
|---------------|-------------|--------|--------------|------|
| Estimation    | GPT-4o-mini | GPT-4o | LLaMA-3.1-8B | Avg  |
| LLaMA-3.2-1B  | 78.0        | 77.6   | 51.4         | 69.0 |
| LLaMA-3.1-8B  | 77.8        | 77.8   | 51.2         | 68.9 |
| Qwen-2.5-7B   | 78.8        | 79.2   | 52.0         | 70.0 |
| LLaMA-3.1-70B | 77.8        | 77.6   | 53.6         | 69.7 |
| (b) GPQA      |             |        |              |      |
| Estimation    | GPT-4o-mini | GPT-4o | LLaMA-3.1-8B | Avg  |
| LLaMA-3.2-1B  | 43.9        | 50.5   | 33.8         | 42.7 |
| LLaMA-3.1-8B  | 41.9        | 51.5   | 35.4         | 42.9 |
| Qwen-2.5-7B   | 41.4        | 50.5   | 34.3         | 42.1 |
| LLaMA-3.1-70B | 42.4        | 53.5   | 34.8         | 43.6 |
| (c) MuSR-ta   |             |        |              |      |
| Estimation    | GPT-4o-mini | GPT-4o | LLaMA-3.1-8B | Avg  |
| LLaMA-3.2-1B  | 83.2        | 90.8   | 80.0         | 84.7 |
| LLaMA-3.1-8B  | 82.8        | 90.4   | 79.6         | 84.3 |
| Qwen-2.5-7B   | 82.0        | 90.8   | 81.6         | 84.8 |
| LLaMA-3.1-70B | 83.6        | 91.2   | 81.6         | 85.5 |

cation. This reduction may suggest a departure from the ostensibly rigorous Bayesian rule. However, previous work on in-context learning has observed that the relative contributions of examples are highly uneven, and that the most relevant examples often account for a disproportionately large

proportion of useful signals (Wang et al., 2024a; Li et al., 2023; Liu et al., 2021). From this perspective, the lightweight version does not discard the conceptual role of the entire backward component, but provides a tractable approximation to its dominant signal. In particular, instead of computing backward term over all  $K$  few-shot examples, ReFeri evaluates only the most relevant exemplar selected by Eq. 9, reducing the backward computation from  $O(K)$  to  $O(1)$  per candidate. Our experiments support this interpretation. We provide detailed comparisons with the *Full* and *No replace* variants in Appendix B.10.

Furthermore, the modularity that separates generation from estimation allows our method to maintain its theoretical validity, even with smaller estimation models. This not only avoids the collapse of Bayesian interpretation, but also provides practical efficiency benefits. To further examine this practical benefit, we provide an extended version of the cost-accuracy analysis from Section 3.3 in Table 22. While the main text reports the summarized trade-off, Table 22 presents full benchmark results. Since all methods select from the same pre-

Table 22: **Full results of cost–accuracy trade-off.** Accuracy and evaluation cost across benchmarks. Costs are measured in actual processing time (seconds) per instance on a single NVIDIA RTX A6000 GPU using the same model configuration. All methods select from the same pre-generated candidate pool, so generation cost is shared and omitted. The best and second-best accuracy scores are highlighted in bold and underline, respectively.

| Size | Methods        | MuSR-ta<br>(Acc. / Time) | MuSR-op<br>(Acc. / Time) | GPQA<br>(Acc. / Time) | MATH500<br>(Acc. / Time) | DROP<br>(Acc. / Time) | HotpotQA<br>(Acc. / Time) | MMLU-PRO<br>(Acc. / Time) | Avg.<br>(Acc. / Time) |
|------|----------------|--------------------------|--------------------------|-----------------------|--------------------------|-----------------------|---------------------------|---------------------------|-----------------------|
| –    | Random         | 76.3 / –                 | 60.8 / –                 | 37.7 / –              | 64.6 / –                 | 72.9 / –              | 32.5 / –                  | 58.5 / –                  | 57.6 / –              |
| 1B   | USC            | 75.5 / 0.80              | <u>62.1</u> / 0.48       | <u>40.9</u> / 0.68    | 65.4 / 0.56              | 73.3 / 0.24           | 32.7 / 0.31               | 58.5 / 0.39               | 58.3 / 0.49           |
|      | USC ( $N=20$ ) | 76.4 / 16.0              | 60.7 / 9.6               | <u>40.9</u> / 13.6    | 64.7 / 11.2              | 74.3 / 4.8            | 32.9 / 6.2                | <u>58.6</u> / 7.8         | 58.4 / 9.8            |
|      | CoT-WP         | 78.0 / 3.06              | 58.9 / 3.06              | <u>40.9</u> / 2.12    | 66.6 / 1.49              | 75.9 / 0.45           | <u>35.3</u> / 0.65        | <u>58.6</u> / 1.39        | 59.2 / 1.75           |
|      | ReFeri         | <b>84.7</b> / 6.10       | <b>62.8</b> / 6.51       | <b>42.7</b> / 4.23    | <b>69.0</b> / 3.12       | <b>77.1</b> / 0.91    | <b>35.9</b> / 1.32        | <b>61.5</b> / 2.78        | <b>62.0</b> / 3.57    |
| 8B   | USC            | 75.6 / 4.67              | <u>61.4</u> / 3.81       | 40.6 / 4.10           | <u>68.3</u> / 3.34       | 76.2 / 2.00           | 35.0 / 2.21               | 61.3 / 2.47               | 59.8 / 3.23           |
|      | USC ( $N=20$ ) | 76.3 / 93.4              | 61.3 / 76.2              | <u>40.9</u> / 82.0    | 68.1 / 66.8              | 76.5 / 40.0           | 35.6 / 44.2               | <b>62.5</b> / 49.4        | 60.2 / 64.6           |
|      | CoT-WP         | <u>79.7</u> / 15.37      | 59.9 / 15.52             | 39.7 / 11.64          | 68.0 / 8.27              | <u>77.1</u> / 2.53    | <b>36.3</b> / 3.70        | 61.6 / 7.64               | <u>60.3</u> / 9.24    |
|      | ReFeri         | <b>84.3</b> / 42.56      | <b>63.7</b> / 42.39      | <b>42.9</b> / 23.64   | <b>68.9</b> / 17.41      | <b>77.4</b> / 6.99    | <u>36.1</u> / 6.94        | <u>61.8</u> / 15.73       | <b>62.2</b> / 22.24   |

Table 23: **Generator–estimator mismatch.** The same 11,020 candidates generated by LLaMA-3.1-8B (excluding MMLU-Pro) are rescored by a matched estimator, a smaller same-family estimator, and a different-family estimator.

| Estimator    | PPL  | corr(PPL, correct.) | ReFeri Acc. | Random |
|--------------|------|---------------------|-------------|--------|
| LLaMA-3.1-8B | 4.19 | −0.411              | 53.1        | 43.7   |
| LLaMA-3.2-1B | 5.25 | −0.395              | 52.4        | 43.7   |
| Qwen-2.5-7B  | 6.20 | −0.396              | <b>53.5</b> | 43.7   |

generated candidate pool, the generation cost is shared across methods; therefore, the reported cost only reflects the additional evaluation time required for response selection.

The extended results show that ReFeri provides a better accuracy-cost trade-off than the baselines. USC with a 1B estimator has the lowest evaluation cost, but its performance remains close to random selection. Increasing the number of USC samples to  $N=20$  with a voting scheme substantially increases computation, yet yields only limited accuracy gains. CoT-WP benefits from a stronger estimator, but it experiences noticeable accuracy degradation when reducing the estimator from 8B to 1B. In contrast, ReFeri maintains stable performance across estimator sizes.

Specifically, ReFeri with a 1B estimator achieves an average accuracy of 62.0%, outperforming 8B CoT-WP, which achieves 60.3%. At the same time, ReFeri with a 1B estimator requires 3.57 seconds per instance on average, compared with 9.24 seconds for the stronger baseline 8B CoT-WP, making it approximately  $2.6\times$  faster.

These results indicate that the lightweight approximation does not collapse the theoretical framework. ReFeri still maintains a conceptual Bayesian structure, and maintains the benefits of backward signal. We believe that this effectiveness, even

when the generation and estimation models are not aligned, is a key strength of our approach. This design choice makes ReFeri broadly applicable and can be easily integrated to existing pipelines.

However, when the estimator differs from the generator, the computed likelihoods no longer reflect the generator’s own distribution, which introduces additional variability. This setting is unavoidable in practice: for closed-source generators the token likelihoods are inaccessible, so verification necessarily relies on a separately chosen estimator. The relevant question is therefore not whether the estimator reproduces the generator’s intrinsic distribution, but whether it still ranks candidate quality effectively. To test this directly, we fix the candidate set and vary only the estimator, rescore the same 11,020 candidates generated by LLaMA-3.1-8B (excluding MMLU-Pro) with a matched estimator (LLaMA-3.1-8B), a smaller same-family estimator (LLaMA-3.2-1B), and a different-family estimator (Qwen-2.5-7B).

As shown in Table 23, absolute likelihoods shift as expected: perplexity rises from 4.19 to 5.25 and then to 6.20, confirming that cross-family mismatch in particular affects calibration. The induced ranking, however, remains stable. The correlation between perplexity and correctness stays near  $-0.40$  in all three cases, and selection accuracy ranges only from 52.4% to 53.5% against a 43.7% random baseline, with the different-family estimator marginally the best despite the worst calibration. This is the property ReFeri relies on, since candidates are ranked within a single query under one fixed estimator and one fixed demonstration set, so estimator-specific offsets are shared across candidates and largely cancel. We note nonetheless that the scores under mismatch express the estimator’s preferences rather than the generator’s own confi-

dence, and can be shaped by calibration, knowledge, reasoning style, verbosity, and tokenization; applicability to closed-source generators therefore rests on the ranking evidence above rather than on the derivation.

### B.10 Additional ablation

Here, we conduct additional experiments to provide comprehensive ablation study for ReFeri. We first evaluate the *Full* variant (Eq. 8), which uses the complete set of examples and therefore provides the most faithful estimate of the backward reconstruction signal (Table 24). However, as discussed in Sec. 2.2, the computational overhead increases linearly with the number of few-shot examples, which renders the *Full* variant less appealing for large-scale or resource-constrained scenarios.

To further analyze this trade-off, we examine the effectiveness of the proposed *prompt replacement* (Eq. 7) for better estimation of backward score. To this end, we consider a simplified variant of our backward reconstruction signal, termed *No replace*, where each few-shot example  $\mathbf{x}_i = (q_i, a_i)$  is evaluated in a one-shot manner using the test query  $\tilde{q}$  and the candidate response  $r_n$  as additional context. Specifically, this variant modifies the backward term in Eq. 8 by replacing the leave-one-out prompt  $\tilde{\mathbf{X}}_i$  with a single pair  $\mathbf{y}_n = (\tilde{q}, r_n)$ :

$$\begin{aligned} S'_{\text{Back}}(\mathbf{y}_n) &:= \log P(\mathbf{X} | \mathbf{y}_n) - \log P(\mathbf{X}) \\ &= \frac{1}{K} \sum_{i=1}^K [\log P(a_i | q_i, \tilde{q}, r_n) - \log P(a_i | q_i)]. \end{aligned} \quad (12)$$

We note that, as in our main method, a cost-efficient variant can be obtained by incorporating the  $i^\dagger$  selection strategy (Eq. 9), which adaptively chooses the most relevant exemplar to the test query.

$$S'_{\text{Back}}(\mathbf{y}_n) := \log P(a_{i^\dagger} | q_{i^\dagger}, \tilde{q}, r_n) - \log P(a_{i^\dagger} | q_{i^\dagger}), \quad (13)$$

This formulation can be interpreted as the most straightforward implementation of backward term (see Eq. 5) under the assumption of mutual independence between few-shot examples. Table 24 reports all four combinations across the three generation models, and two observations follow. First, prompt replacement is consistently beneficial: ReFeri improves over *No replace* on every generator (71.1 vs. 71.0, 63.4 vs. 63.1, and 51.9 vs. 49.8), and the same holds between their *Full* counterparts. Second, ReFeri is robust to how many exemplars enter the backward term, matching the *Full* variant within 0.4 points on all three generators while re-

quiring only  $O(1)$  computation per candidate. This indicates that *No replace* still provides a useful backward penalty, but the penalty is less contextualized because it reconstructs each few-shot example using only the candidate pair  $\mathbf{y}_n$ . Nonetheless, *No replace* could serve as a practical alternative that trades off a small performance drop with the greater simplicity.

Table 25 further extends the component ablation in Table 3 to all three generation models and all seven tasks. In addition to the *Direct* and *Forward* variants, we consider two single-term variants that select candidates using only the backward score, either by choosing the largest (*Backward*) or the smallest (*-Backward*) value. As shown in Table 25, neither variant performs competitively across the three generators, suggesting that the backward score alone does not serve as a reliable estimate of response correctness. We also observe that *Direct* and *Forward* exhibit complementary behavior across tasks, with neither consistently outperforming the other. In contrast, ReFeri combines the two signals and is never worse than both variants across the 21 generator–task combinations, while achieving the best average accuracy for all three generators. The improvement over the forward-only variant is also consistent across the three matched runs, as further supported by the significance analysis in Table 8.

### B.11 Additional comparison with direct scoring

To further verify the concern raised in Section 2.2 that directly scoring candidate is unreliable when the model is miscalibrated in unfamiliar domains, we perform an additional comparison by fixing the candidate set and varying only the input condition. We reuse the same  $N$  candidate responses  $r_n$  generated from Table 1 under few-shot prompting  $(\tilde{q}, \mathbf{X})$ . The results in Table 26 were calculated by averaging the three independent sampling runs used and consolidating the modeled averages into a single table. For *direct confidence score*, we estimate each candidate the same as Eq. 6, but without few-shot examples:

$$S_{\text{Direct}}(r_n) := \log P(r_n | \tilde{q}). \quad (14)$$

We also consider *Self-Certainty (zero)*, a zero-shot conditioning variant of Self-Certainty, where uncertainty is computed without demonstrations. Unlike the main table, we replace the conditioning  $(\tilde{q}, \mathbf{X})$

Table 24: **Ablation study on backward terms.** We compare the proposed leave-one-out replacement strategy with a simpler no-replacement variant, across all three generation models. Full variants average the backward signal over all few-shot examples, while lightweight variants use a single selected exemplar. The best average within each generator is in **bold**.

| Models       | Methods           | MuSR-ta<br>(Acc.) | MuSR-op<br>(Acc.) | GPQA<br>(Acc.) | MATH500<br>(Acc.) | DROP<br>(EM / F1)         | HotpotQA<br>(EM / F1)     | MMLU-PRO<br>(Acc.) | Avg.        |
|--------------|-------------------|-------------------|-------------------|----------------|-------------------|---------------------------|---------------------------|--------------------|-------------|
| GPT-4o       | No replace (full) | 90.4              | <b>74.2</b>       | <b>52.0</b>    | 78.0              | 83.4 / 90.8               | 45.6 / 59.6               | 75.3               | <b>71.3</b> |
|              | No replace        | <b>90.8</b>       | 72.3              | 50.5           | 77.8              | <b>83.6</b> / 90.9        | <b>47.2</b> / <b>61.1</b> | 75.1               | 71.0        |
|              | ReFeri (Full)     | <b>90.8</b>       | 70.7              | 51.0           | <b>78.4</b>       | <b>83.6</b> / <b>91.2</b> | 47.0 / 60.6               | <b>75.5</b>        | 71.0        |
|              | ReFeri            | 90.4              | 71.9              | 51.5           | 77.8              | <b>83.6</b> / 91.1        | 47.0 / 60.7               | 75.4               | 71.1        |
| GPT-4o-mini  | No replace (full) | <b>82.8</b>       | 60.2              | 42.4           | <b>78.0</b>       | 78.4 / 84.2               | <b>36.2</b> / <b>48.0</b> | <b>65.0</b>        | 63.3        |
|              | No replace        | 82.4              | 60.2              | <b>42.9</b>    | 77.6              | 78.4 / 84.1               | 35.8 / 47.6               | 64.7               | 63.1        |
|              | ReFeri (Full)     | <b>82.8</b>       | <b>61.3</b>       | 42.4           | 77.8              | <b>79.6</b> / <b>85.3</b> | 35.8 / 47.9               | <b>65.0</b>        | <b>63.5</b> |
|              | ReFeri            | <b>82.8</b>       | <b>61.3</b>       | 41.9           | 77.8              | 79.2 / 84.9               | <b>36.2</b> / <b>48.0</b> | 64.9               | 63.4        |
| LLaMA-3.1-8B | No replace (full) | 73.2              | 55.5              | 34.8           | 50.8              | 67.6 / 73.4               | 24.6 / 32.7               | 44.5               | 50.2        |
|              | No replace        | 72.4              | 55.9              | 33.8           | 50.8              | 66.2 / 72.7               | 24.6 / 32.6               | 44.7               | 49.8        |
|              | ReFeri (Full)     | <b>79.6</b>       | 55.9              | 34.8           | 50.8              | <b>70.2</b> / <b>76.4</b> | 24.6 / 32.8               | 44.9               | 51.5        |
|              | ReFeri            | <b>79.6</b>       | <b>57.8</b>       | <b>35.4</b>    | <b>51.2</b>       | 69.4 / 75.7               | <b>25.0</b> / <b>33.2</b> | <b>45.1</b>        | <b>51.9</b> |

Table 25: **Extended ablation across all generators and tasks.** Each row uses the same candidate pool and differs only in the selection score, averaged over three runs. *Backward* and *−Backward* select the largest and the smallest raw backward score, and *ReFeri (Full)* uses the full backward term of Eq. 8. The best score within each generator is in **bold**.

| Models       | Methods       | MuSR-ta<br>(Acc.)     | MuSR-op<br>(Acc.)     | GPQA<br>(Acc.)        | MATH500<br>(Acc.)     | DROP<br>(EM / F1)                             | HotpotQA<br>(EM / F1)                         | MMLU-PRO<br>(Acc.)    | Avg.        |
|--------------|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------------------------------|-----------------------------------------------|-----------------------|-------------|
| GPT-4o       | Backward      | 87.1 $\pm$ 0.2        | 70.7 $\pm$ 2.3        | 47.6 $\pm$ 1.9        | 75.3 $\pm$ 0.7        | 80.9 $\pm$ 0.9 / 89.3 $\pm$ 0.9               | 45.7 $\pm$ 0.4 / 59.1 $\pm$ 0.4               | 73.4 $\pm$ 0.3        | 68.7        |
|              | −Backward     | 87.5 $\pm$ 0.7        | 69.7 $\pm$ 1.4        | 48.3 $\pm$ 1.7        | 75.0 $\pm$ 0.2        | 78.7 $\pm$ 0.6 / 88.1 $\pm$ 0.5               | 45.7 $\pm$ 1.2 / 59.1 $\pm$ 0.7               | 74.0 $\pm$ 0.2        | 68.4        |
|              | Direct        | 90.8 $\pm$ 1.1        | 73.3 $\pm$ 2.4        | <b>52.2</b> $\pm$ 1.3 | 77.2 $\pm$ 1.7        | 81.9 $\pm$ 2.0 / 89.8 $\pm$ 1.2               | 46.7 $\pm$ 0.5 / 60.5 $\pm$ 0.8               | 75.4 $\pm$ 0.1        | 71.1        |
|              | Forward       | 90.8 $\pm$ 0.4        | 73.5 $\pm$ 2.8        | 51.5 $\pm$ 0.5        | 78.4 $\pm$ 0.5        | 83.1 $\pm$ 0.3 / 90.6 $\pm$ 0.5               | <b>47.2</b> $\pm$ 0.2 / <b>61.4</b> $\pm$ 0.4 | 75.4 $\pm$ 0.0        | 71.4        |
|              | ReFeri        | 90.8 $\pm$ 0.4        | <b>73.7</b> $\pm$ 2.2 | 51.8 $\pm$ 0.3        | 78.5 $\pm$ 0.6        | <b>83.5</b> $\pm$ 0.1 / <b>90.9</b> $\pm$ 0.6 | 46.9 $\pm$ 0.4 / 61.0 $\pm$ 0.7               | 75.4 $\pm$ 0.3        | <b>71.5</b> |
|              | ReFeri (Full) | <b>90.9</b> $\pm$ 0.2 | 73.0 $\pm$ 2.1        | 51.2 $\pm$ 0.2        | <b>78.7</b> $\pm$ 0.2 | 83.3 $\pm$ 0.4 / 90.8 $\pm$ 0.5               | 47.1 $\pm$ 0.2 / 61.3 $\pm$ 0.5               | 75.4 $\pm$ 0.0        | 71.4        |
| GPT-4o-mini  | Backward      | 78.7 $\pm$ 0.7        | 61.6 $\pm$ 2.3        | 40.9 $\pm$ 1.6        | 74.1 $\pm$ 0.7        | 77.3 $\pm$ 0.8 / 83.5 $\pm$ 0.5               | 34.3 $\pm$ 0.7 / 45.5 $\pm$ 0.6               | 62.7 $\pm$ 0.5        | 61.4        |
|              | −Backward     | 76.0 $\pm$ 1.6        | 60.2 $\pm$ 1.7        | 41.4 $\pm$ 2.3        | 74.5 $\pm$ 0.7        | 75.5 $\pm$ 0.5 / 82.3 $\pm$ 0.4               | 33.6 $\pm$ 0.3 / 44.9 $\pm$ 0.2               | 62.8 $\pm$ 0.0        | 60.6        |
|              | Direct        | 81.3 $\pm$ 0.8        | 62.1 $\pm$ 2.1        | 44.3 $\pm$ 1.3        | <b>78.3</b> $\pm$ 0.2 | 78.9 $\pm$ 0.6 / 84.2 $\pm$ 0.9               | 35.1 $\pm$ 0.6 / 46.9 $\pm$ 0.4               | 64.8 $\pm$ 0.2        | 63.5        |
|              | Forward       | 82.9 $\pm$ 0.2        | 61.8 $\pm$ 0.8        | 44.3 $\pm$ 2.3        | 78.1 $\pm$ 0.3        | 79.1 $\pm$ 0.4 / 84.6 $\pm$ 0.7               | 35.3 $\pm$ 0.5 / <b>47.4</b> $\pm$ 0.6        | 64.8 $\pm$ 0.2        | 63.8        |
|              | ReFeri        | <b>83.1</b> $\pm$ 0.2 | 62.0 $\pm$ 0.6        | <b>44.6</b> $\pm$ 2.6 | 78.2 $\pm$ 0.4        | 79.1 $\pm$ 0.5 / <b>84.7</b> $\pm$ 0.8        | <b>35.6</b> $\pm$ 0.5 / <b>47.4</b> $\pm$ 0.6 | 64.9 $\pm$ 0.4        | <b>63.9</b> |
|              | ReFeri (Full) | 82.8 $\pm$ 0.3        | <b>62.2</b> $\pm$ 0.7 | 44.1 $\pm$ 1.3        | 78.1 $\pm$ 0.2        | 79.1 $\pm$ 0.7 / 84.6 $\pm$ 0.9               | 35.4 $\pm$ 0.3 / <b>47.4</b> $\pm$ 0.4        | <b>65.0</b> $\pm$ 0.2 | 63.8        |
| LLaMA-3.1-8B | Backward      | 65.1 $\pm$ 0.9        | 51.6 $\pm$ 0.8        | 22.4 $\pm$ 2.1        | 41.0 $\pm$ 1.6        | 64.7 $\pm$ 0.9 / 69.5 $\pm$ 0.9               | 19.5 $\pm$ 1.9 / 25.8 $\pm$ 1.9               | 36.9 $\pm$ 0.3        | 43.0        |
|              | −Backward     | 61.5 $\pm$ 2.4        | 51.8 $\pm$ 2.4        | 25.9 $\pm$ 2.3        | 44.3 $\pm$ 1.3        | 50.8 $\pm$ 1.5 / 56.6 $\pm$ 1.3               | 17.7 $\pm$ 0.7 / 22.8 $\pm$ 1.2               | 40.4 $\pm$ 0.3        | 41.8        |
|              | Direct        | 68.4 $\pm$ 3.0        | 56.0 $\pm$ 4.8        | 32.6 $\pm$ 2.0        | <b>50.3</b> $\pm$ 1.3 | 68.2 $\pm$ 0.0 / 74.0 $\pm$ 0.1               | 25.1 $\pm$ 1.0 / 33.2 $\pm$ 1.5               | 44.0 $\pm$ 0.3        | 49.2        |
|              | Forward       | <b>76.9</b> $\pm$ 4.6 | 57.0 $\pm$ 0.4        | 32.3 $\pm$ 1.3        | 50.1 $\pm$ 1.6        | <b>71.5</b> $\pm$ 1.7 / <b>76.7</b> $\pm$ 0.5 | 25.6 $\pm$ 0.8 / <b>33.6</b> $\pm$ 0.7        | 44.6 $\pm$ 0.2        | 51.1        |
|              | ReFeri        | 76.5 $\pm$ 3.4        | <b>57.5</b> $\pm$ 0.5 | <b>33.5</b> $\pm$ 1.7 | 50.2 $\pm$ 1.6        | 70.9 $\pm$ 1.6 / 76.3 $\pm$ 0.5               | <b>25.7</b> $\pm$ 0.7 / <b>33.6</b> $\pm$ 0.6 | <b>44.8</b> $\pm$ 0.3 | <b>51.3</b> |
|              | ReFeri (Full) | 76.8 $\pm$ 2.0        | 56.5 $\pm$ 0.5        | 32.7 $\pm$ 1.7        | 50.1 $\pm$ 1.2        | 70.9 $\pm$ 1.2 / 76.3 $\pm$ 0.4               | 25.6 $\pm$ 0.8 / 33.5 $\pm$ 0.7               | 44.6 $\pm$ 0.3        | 51.0        |

with ( $\tilde{q}$ ) when computing the uncertainty for each  $r_n$ .

As shown in Table 26, candidates that score directly without demonstrations can be competitive in relatively familiar tasks such as MATH500, but become unreliable as the domain shifts. In particular, the largest gap appears in the MuSR-ta. ReFeri improves by 3.3% (83.5 vs. 80.2) over the direct score. This demonstrates that relying solely on internal model knowledge is unreliable in unfamiliar domains. We also observe significant performance drop with Self-Certainty (zero). Self-Certainty quantifies the uncertainty by measuring how close the model’s next token distribution is

to the uniform distribution. When eliminating a few-shot examples, Self-Certainty can make the predictive distribution flat, especially in unfamiliar domain with weak calibration. In this case, it may not provide sufficient signals to select a plausible candidate among the candidates. On the other hand, ReFeri consistently benefits from recycling a few-shot context during validation, which provides external grounding signals and improves reliability, especially on benchmarks where few-shot guidance is important.

## B.12 Detailed analysis of the backward term

In this subsection, we provide the details of the analyses summarized in Section 3.3. As described

Table 26: **Additional comparison to direct terms.** Model-averaged results comparing ReFeri with direct likelihood scoring and zero-shot variants of uncertainty-based selection. All methods use the same candidate responses, but differ in whether the estimator is conditioned on few-shot examples.

| Methods               | MuSR-ta<br>(Acc.) | MuSR-op<br>(Acc.) | GPQA<br>(Acc.) | MATH500<br>(Acc.) | DROP<br>(EM / F1) | HotpotQA<br>(EM / F1) | Avg.        |
|-----------------------|-------------------|-------------------|----------------|-------------------|-------------------|-----------------------|-------------|
| Direct                | <u>80.2</u>       | <u>63.8</u>       | <u>43.0</u>    | <u>68.6</u>       | <u>76.3</u>       | <u>35.6</u>           | <u>61.2</u> |
| Self-Certainty (zero) | 79.2              | 62.5              | 40.8           | 68.2              | 76.2              | 34.7                  | 60.3        |
| <b>ReFeri</b>         | <b>83.5</b>       | <b>64.4</b>       | <b>43.3</b>    | <b>69.0</b>       | <b>77.8</b>       | <b>36.1</b>           | <b>62.4</b> |

Table 27: **Agreement between forward and backward selections.** We measure how often the candidate preferred by the forward confidence score matches the candidate with the largest raw backward reconstruction score.

| Models          | Agreement | MuSR-ta | MuSR-op | GPQA  | MATH500 | DROP  | HotpotQA | MMLU-PRO | Avg   |
|-----------------|-----------|---------|---------|-------|---------|-------|----------|----------|-------|
| GPT-4o-mini     | (F,B)     | 0.188   | 0.180   | 0.242 | 0.242   | 0.234 | 0.244    | 0.208    | 0.220 |
| GPT-4o          | (F,B)     | 0.212   | 0.238   | 0.303 | 0.222   | 0.268 | 0.260    | 0.226    | 0.247 |
| Llama-3.1-8B-It | (F,B)     | 0.140   | 0.312   | 0.066 | 0.182   | 0.246 | 0.226    | 0.143    | 0.188 |

Table 28: **What the raw backward score correlates with.** Candidate-level Pearson correlations with correctness, response length, and similarity to the retrieved demonstration answer, measured by stemmed ROUGE-L and all-mpnet-base-v2 cosine. *Replace* is the score used by ReFeri (Eq. 10) and *No-replace* the diagnostic variant of Eq. 13.

| Task     | Correct. | Length | Replace |        | No-replace |        |
|----------|----------|--------|---------|--------|------------|--------|
|          |          |        | ROUGE-L | cosine | ROUGE-L    | cosine |
| MuSR-ta  | -0.011   | 0.133  | -0.028  | 0.041  | 0.758      | 0.578  |
| MuSR-op  | -0.049   | -0.043 | 0.311   | 0.126  | 0.638      | 0.296  |
| GPQA     | -0.029   | 0.114  | 0.046   | 0.001  | 0.038      | 0.037  |
| MATH500  | -0.034   | 0.056  | -0.026  | -0.013 | 0.052      | 0.001  |
| DROP     | 0.088    | -0.225 | 0.302   | 0.078  | 0.455      | 0.365  |
| HotpotQA | -0.012   | -0.012 | 0.120   | 0.091  | 0.424      | -0.016 |
| MMLU-Pro | 0.030    | -0.117 | 0.132   | 0.049  | 0.364      | 0.165  |

in Section 2.2, the forward confidence score measures the likelihood of a candidate given the few-shot examples, while the backward reconstruction penalty suppresses candidates that are overly specific to the demonstrations. Intuitively, a large backward term implies that the candidate mimics the surface patterns or artifacts of the few-shot examples rather than reflecting the underlying reasoning logic. However, the same term also increases when a candidate correctly reuses a reasoning path that the demonstrations instantiate, and both cases occur in practice. The role of the backward term therefore cannot be determined from its magnitude alone. To this end, we examine what the raw score does not capture, what it actually tracks, and whether this sensitivity changes the selected response.

We first verify that the raw backward score is not a redundant estimate of response quality. As shown in Table 27, the candidate preferred by the forward score coincides with the one receiving the largest raw backward score in only 0.22 of the

cases on average; since  $N=5$ , chance agreement is approximately 0.2, and the two terms therefore rank candidates almost independently. Table 28 supports the same conclusion at the candidate level. The correlation with correctness is close to zero on all seven tasks ( $|r| \leq 0.09$ ) and is as often negative as positive. If subtracting the backward term mainly penalized candidates that are simply better task instances, this correlation should be clearly positive. The correlation with response length is also limited ( $|r| \leq 0.23$ , with most values below 0.15), showing that the signal is not a proxy for verbosity.

We next examine what the raw score actually tracks. As shown in Table 28, the replacement score used by ReFeri is positively associated with the lexical similarity between the candidate and the retrieved demonstration answer on most tasks. This association becomes substantially stronger under the no-replacement variant of Eq. 13 (up to 0.758 lexical and 0.578 semantic on MuSR-ta), where the contribution of the candidate is more directly isolated. This indicates that the raw backward term captures proximity to the demonstrations rather than response quality, which is why it serves as a penalty only after the forward score has established that a candidate is plausible.

The perturbation experiment in Figure 4 indicates that this sensitivity is stronger for the backward term, and Table 29 reports the same measurements with 95% confidence intervals, where the increase in the backward term exceeds that of the forward score in every generator-task cell. We then examine whether this difference changes the selected response, by injecting a demonstration artifact into a candidate that is already known to be

Table 29: **Perturbation with demonstration fragments.** Raw forward and backward scores before and after prepending the retrieved demonstration answer, with 95% confidence intervals; the tabular counterpart of Figure 4. The change is larger for the backward term in every cell.

| Task                  | Generator    | Original                | $\rho=100$              | $\Delta$      |
|-----------------------|--------------|-------------------------|-------------------------|---------------|
| <i>Forward score</i>  |              |                         |                         |               |
| MATH500               | LLaMA-3.1-8B | -0.677 [-0.722, -0.634] | -0.544 [-0.575, -0.515] | +0.133        |
|                       | GPT-4o-mini  | -0.668 [-0.689, -0.648] | -0.509 [-0.526, -0.493] | +0.159        |
|                       | GPT-4o       | -0.744 [-0.777, -0.712] | -0.551 [-0.577, -0.527] | +0.193        |
| MuSR-ta               | LLaMA-3.1-8B | -1.550 [-1.640, -1.460] | -0.831 [-0.870, -0.793] | +0.719        |
|                       | GPT-4o-mini  | -1.236 [-1.250, -1.222] | -0.766 [-0.775, -0.758] | +0.470        |
|                       | GPT-4o       | -1.180 [-1.195, -1.165] | -0.709 [-0.718, -0.700] | +0.471        |
| <i>Backward score</i> |              |                         |                         |               |
| MATH500               | LLaMA-3.1-8B | 0.092 [0.092, 0.092]    | 0.396 [0.391, 0.401]    | <b>+0.304</b> |
|                       | GPT-4o-mini  | 0.227 [0.227, 0.228]    | 1.002 [0.985, 1.021]    | <b>+0.775</b> |
|                       | GPT-4o       | 0.227 [0.227, 0.228]    | 1.002 [0.984, 1.021]    | <b>+0.775</b> |
| MuSR-ta               | LLaMA-3.1-8B | 0.641 [0.640, 0.642]    | 1.482 [1.453, 1.512]    | <b>+0.841</b> |
|                       | GPT-4o-mini  | 0.602 [0.602, 0.603]    | 1.536 [1.511, 1.561]    | <b>+0.934</b> |
|                       | GPT-4o       | 0.601 [0.601, 0.601]    | 1.536 [1.510, 1.561]    | <b>+0.935</b> |

incorrect. For each test query, we reuse the five candidates generated in the first run of Table 1 and retrieve the demonstration with Eq. 9, so that all five candidates of a query share the same exemplar. Among the incorrect candidates, we inject the artifact into the one with the *lowest* forward score rather than one that the forward score already favors, which makes the test conservative: the artifact must be strong enough to overturn a selection that was initially unlikely. Concretely, we take the answer text of the retrieved demonstration, tokenize it with the tokenizer of the estimator in use, and prepend its first  $\rho\%$  of tokens to that candidate, with  $\rho \in \{0, 25, 50, 100\}$ . Only the modified candidate is rescored, so  $\rho=0$  reproduces the original selection, and we keep the correctness labels fixed at their pre-injection values so that any change is attributable to the selector. This leaves 1,687 candidate sets pooled from three generators and three tasks.

As shown in Table 4, the resulting pattern is monotone. Forward-only accuracy degrades from 45.29% to 33.31% with the 8B estimator and from 45.41% to 30.71% with the 1B estimator, since the injected prefix makes an incorrect candidate appear more consistent with the demonstrations. In contrast, ReFeri improves to 47.48% and 47.66%, because the same insertion inflates the backward term more than the forward term and is subtracted away. Notably, the two estimator scales behave almost identically, indicating that this behavior stems from the scoring design rather than from estimator capacity.

Finally, we turn from an injected dependence to the demonstration similarity that arises naturally in

the candidate pool. Here, we modify nothing and reuse the candidates and 8B estimator scores from the first run of Table 1. Within each query that contains both correct and incorrect candidates, we take the strongest candidate of either kind under the forward score,  $c^+$  and  $c^-$ , so that the two most competitive candidates are directly opposed. For both, we measure the ROUGE-L similarity against the retrieved demonstration answer  $a_{i\ddagger}$  and sort their difference within each generator $\times$ task cell, comparing the two extreme quartiles. These correspond to the cases where the correct and the incorrect candidate is far more demonstration-similar, and are consistent with the reuse of a shared reasoning path and with demonstration mimicry, respectively.

Writing  $\Delta B = \tilde{S}_{\text{Back}}(c^-) - \tilde{S}_{\text{Back}}(c^+)$ , the margin by which ReFeri prefers the correct candidate decomposes as

$$\underbrace{S_{\text{Forw}}(c^+) - S_{\text{Forw}}(c^-)}_{\text{forward margin}} + \underbrace{\Delta B}_{\text{backward correction}}, \quad (15)$$

so that the penalty helps exactly when the incorrect candidate receives the larger reconstruction score. As shown in Table 5, both cases occur. When the correct candidate is more demonstration-similar, it does receive the larger backward score, confirming that reconstruction can reflect a correctly reused reasoning path; subtracting it nevertheless reduces pair accuracy by only 0.63 points, since that candidate also retains strong forward support. When the incorrect candidate is more similar, it receives the larger penalty and ReFeri improves by 2.23 points. The penalty therefore remains close to inert when demonstration similarity is benign and becomes useful when it is not, which is how subtracting

it discounts candidates whose forward support is overly explained by the few-shot examples.

## C Qualitative Examples

In this section, we present qualitative examples to further analyze the proposed ReFeri. For better readability, we only present the examples from MATH500, GPQA, and HotpotQA. All the responses are generated by GPT-4o-mini, and we use the ReFeri (*Full*) variant for illustration to provide the clearest comparisons.

### C.1 Token level analysis

To better understand how ReFeri identifies high-quality response using given few-shot examples, we perform a token-level analysis of following backward reconstruction penalty (Eq. 8). The visualization results are shown in Table 30–Table 32.

For a given test query  $\tilde{q}$ , we divide the candidate responses into correct and incorrect groups using ground-truth labels. For each token, we calculate the difference between its token-level backward score under the two groups. When the backward term for the tokens in the few-shot examples exhibit lower score in the correct group compared to incorrect one, the tokens are colored *red*. In other case, the tokens are colored *blue*. For visual clarity, we only highlight the top 60% of tokens based on the absolute difference in values. The remaining 40% remain uncolored. The value in parentheses is the ratio of tokens highlighted in red to the total number of tokens.

Since ReFeri subtracts the backward term from the forward score, lower token-level backward scores for the correct candidate represent the token whose backward term contributes to distinguishing plausible responses. Thus, the red token can be interpreted as part of several examples in which the right candidate induces less demonstration-specific reconstruction than the wrong candidate. For example, in MATH500 and GPQA, numbers, symbols, final-answer formats, and intermediate reasoning steps are often highlighted in red, suggesting that the backward term helps identify candidates that avoid merely reproducing demonstration-specific structures. In HotpotQA can also see that red is dominant for most, but not all, meaningful words.

### C.2 Response level analysis

For each response selected by ReFeri, we compute the Eq. 11 for both best and worst response. The

value in parentheses is the final selection score. The highest score among all candidates corresponds to the best response, and the lowest score represents the worst response. The visualization results are shown in Table 33–Table 35.

As shown in below examples, the selected response by ReFeri has more accurate reasoning. For instance, in the example of MATH500, the best-scoring response anchors their reasoning in a coordinate system and follow clean logic. The worst response, on the other hand, ignore spatial cues, misapply subtraction, and over-complicate with lengthy and internally inconsistent steps. Similarly, in GPQA, the best response reasonably combines the results from the two analyses to arrive at the correct answer. The worst response, on the other hand, appears to rely primarily on the idea that “*the heavy branching in (A) is consistent with the splitting observed*” without making a clear connection to the analyzed results. Thus, one response integrates the data, while the other reduces it to a vague notion of complexity. In HotpotQA, selected answer is consistent with HotpotQA’s multi-hop requirements by making intermediate hops explicit and factually correct, while rejected answer provides unsupported single-hop claims.

## D Usage of AI Assistants

This paper used AI-based writing aids to improve sentence structure, correct grammar, and improve readability. These tools were applied only to language refinement and did not affect the advancement of technical content, research methodology, or experimental analysis. All scientific ideas, results, and conclusions were conceived and written entirely by researchers. The use of AI aids was limited to editorial purposes and did not impair the originality or intellectual contribution of the work.

Table 30: Token-level qualitative analysis on MATH500.

---

Few-shot examples for MATH500 (0.6455):

---

**Test-Query:**

Twelve 1 by 1 squares form a rectangle, as shown. What is the total area of the shaded region? [asy]

```
size(4cm);defaultpen(linewidth(0.75));

// Filled portions
fill((0, 4)–(0, 0)–(2, 0)–cycle, gray(0.75));
fill((0, 4)–(3, 4)–(3, 0)–cycle, gray(0.75));

// grid
int j;
for (j = 0; j < 4; ++j) draw((j, 0)–(j, 4));
for (j = 0; j < 5; ++j) draw((0, j)–(3, j));

//diagonals
draw((0, 4)–(3, 0)); draw((0, 4)–(2, 0));
[/asy]
```

---

**Few-shot 1 Query:**

Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$  of the remaining distance. How far has he hopped after five hops? Express your answer as a common fraction.

**Few-shot 1 Response:**

Let's think step by step  
 Kevin hops  $\frac{1}{3}$  of the remaining distance with every hop.  
 His first hop takes  $\frac{1}{3}$  closer.  
 For his second hop, he has  $\frac{2}{3}$  left to travel, so he hops forward  $\frac{2}{3}(\frac{1}{3})$ .  
 For his third hop, he has  $(\frac{2}{3})^2$  left to travel, so he hops forward  $(\frac{2}{3})^2(\frac{1}{3})$ .  
 In general, Kevin hops forward  $(\frac{2}{3})^{k-1}(\frac{1}{3})$  on his  $k$ th hop.  
 We want to find how far he has hopped after five hops.  
 This is a finite geometric series with first term  $\frac{1}{3}$ , common ratio  $\frac{2}{3}$ , and five terms.  
 Thus, Kevin has hopped  $\frac{\frac{1}{3}(1-(\frac{2}{3})^5)}{1-\frac{2}{3}} = \boxed{\frac{211}{243}}$ .  
 The answer is  $\frac{211}{243}$

---

**Few-shot 2 Query:**

What is the area of the region defined by the equation  $x^2 + y^2 - 7 = 4y - 14x + 3$ ?

**Few-shot 2 Response:**

Let's think step by step  
 We rewrite the equation as  $x^2 + 14x + y^2 - 4y = 10$  and then complete the square,  
 resulting in  $(x+7)^2 - 49 + (y-2)^2 - 4 = 10$ ,  
 or  $(x+7)^2 + (y-2)^2 = 63$ .  
 This is the equation of a circle with center  $(-7, 2)$  and radius  $\sqrt{63}$ ,  
 so the area of this region is  $\pi r^2 = \boxed{63\pi}$ .  
 The answer is  $63\pi$

---

**Few-shot 3 Query:**

If  $x^2 + y^2 = 1$ , what is the largest possible value of  $|x| + |y|$ ?

**Few-shot 3 Response:**

Let's think step by step  
 If  $(x, y)$  lies on the circle,  
 so does  $(x, -y)$ ,  $(-x, -y)$ , and  $(-x, y)$ , (which all give the same value of  $|x| + |y|$ ),  
 so we can assume that  $x \geq 0$  and  $y \geq 0$ . Then  $|x| + |y| = x + y$ . Squaring, we get  $(x + y)^2 = x^2 + 2xy + y^2 = 1 + 2xy$ .  
 Note that  $(x - y)^2 \geq 0$ . Expanding, we get  $x^2 - 2xy + y^2 \geq 0$ , so  $2xy \leq x^2 + y^2 = 1$ . Hence,  $1 + 2xy \leq 2$ , which means  $x + y \leq \sqrt{2}$ . Equality occurs when  $x = y = \frac{1}{\sqrt{2}}$ ,  
 so the maximum value of  $|x| + |y|$  is  $\boxed{\sqrt{2}}$ .  
 The answer is  $\sqrt{2}$

---

---

Few-shot examples for MATH500 (0.6455):

---

**Few-shot 4 Query:**

If  $f(x) = \frac{ax+b}{cx+d}$ ,  $abcd \neq 0$  and  $f(f(x)) = x$  for all  $x$  in the domain of  $f$ , what is the value of  $a + d$ ?

**Few-shot 4 Response:**

Let's think step by step

The condition  $f(f(x)) = x$  means that  $f$  is the inverse of itself,

so its graph is symmetrical about the line  $y = x$ .

With a rational function of this form, we will have two asymptotes:

a vertical one at  $x = -d/c$  if  $cx+d$  does not divide  $ax+b$ ,

and a horizontal one at  $y = a/c$ ,

if we take the limit of  $f(x)$  as  $x$  goes to  $\pm\infty$ .

In order for  $f$  to be its own inverse, the intersection of the asymptotes must lie on the line  $y = x$  so that it and its asymptotes reflect onto themselves.

This means that  $-d/c = a/c$ ,

and therefore  $-d = a$  and  $a + d = \boxed{0}$ .

The answer is 0

---

**Few-shot 5 Query:**

Expand  $(2z^2 + 5z - 6)(3z^3 - 2z + 1)$ .

**Few-shot 5 Response:**

Let's think step by step

```
$$\begin{array}{l}
```

```
& & 3z^3 & & -2z & + 1 & \\
```

```
& \times & & 2z^2 & + 5z & - 6 & \\
```

```
\cline{1-7}\rule{0pt}{0.17in}
```

```
& & -18z^3 & + 12z & - 6 & \\
```

```
& + 15z^4 & - 10z^2 & + 5z & \\
```

```
+ & 6z^5 & - 4z^3 & + 2z^2 & \\
```

```
\cline{1-7}\rule{0pt}{0.17in}
```

```
& 6z^5 & + 15z^4 & - 22z^3 & - 8z^2 & + 17z & - 6 & \\
```

```
\end{array}$$$ The answer is  $6z^5 + 15z^4 - 22z^3 - 8z^2 + 17z - 6$ 
```

---

Table 31: Token-level qualitative analysis on GPQA.

---

Few-shot examples for GPQA (0.6970):

---

**Test-Query:**

A chemist performs two reactions:

Reaction 1: (E)-oct-4-ene is treated with one equiv. of mCPBA, followed by aqueous acid.

Reaction 2: (Z)-oct-4-ene is treated with one equiv. of mCPBA, followed by aqueous acid.

Both reactions go to 100% completion. The chemist combines the products of both reactions, and then runs the product mixture on both a standard (achiral) reverse-phase HPLC column and a chiral HPLC column.

Assuming that the chromatographic resolution is as high as theoretically possible in both cases, what does the chemist observe in each of these chromatograms?

---

**Few-shot 1 Query:**

In a given population, 1 out of every 400 people has a cancer caused by a completely recessive allele,  $b$ . Assuming the population is in Hardy-Weinberg equilibrium, which of the following is the expected proportion of individuals who carry the  $b$  allele but are not expected to develop the cancer?

**Few-shot 1 Response:**

The expected proportion of individuals who carry the  $b$  allele but are not expected to develop the cancer equals to the frequency of heterozygous allele in the given population.

According to the Hardy-Weinberg equation  $p^2 + 2pq + q^2 = 1$ , where  $p$  is the frequency of dominant allele frequency,  $q$  is the frequency of recessive allele frequency,  $p^2$  is the frequency of the homozygous dominant allele,  $q^2$  is the frequency of the recessive allele, and  $2pq$  is the frequency of the heterozygous allele.

Given that  $q^2 = 1/400$ , hence,  $q = 0.05$  and  $p = 1 - q = 0.95$ .

The frequency of the heterozygous allele is  $2pq = 2 \times 0.05 \times 0.95 = 38/400$ .

---

---

### Few-shot examples for GPQA (0.6970):

---

#### Few-shot 2 Query:

A Fe pellet of 0.056 g is first dissolved in 10 mL of hydrobromic acid HBr (0.1 M). The resulting solution is then titrated by KMnO<sub>4</sub> (0.02 M). How many equivalence points are there?

#### Few-shot 2 Response:

HBr reacts with Fe to produce Fe<sup>2+</sup>.

MnO<sub>4</sub><sup>-</sup> initially reacts with Fe<sup>2+</sup> followed by Br<sup>-</sup>.

There are two equivalence points at 25 mL and 35 mL.

In the beaker, the present species are Fe<sup>2+</sup> and Br<sup>-</sup>.

In a titration involving two analytes, it's essential to identify which reaction occurs first.

Given the redox nature of the titration and the reduction potentials: E<sup>0</sup> (Br<sub>2</sub>/Br<sup>-</sup>) = 1.09 V, E<sup>0</sup> (MnO<sub>4</sub><sup>-</sup>/Mn<sup>2+</sup>) = 1.49 V, and E<sup>0</sup> (Fe<sup>3+</sup>/Fe<sup>2+</sup>) = 0.77 V.

With [Fe<sup>2+</sup>] determined as 0.1M, two reactions are considered.

Reaction 1: MnO<sub>4</sub><sup>-</sup> reacts with 5Fe<sup>2+</sup> and 8H<sup>+</sup> to produce Mn<sup>2+</sup>, 5Fe<sup>3+</sup>, and 4H<sub>2</sub>O.

Reaction 2: 2MnO<sub>4</sub><sup>-</sup> reacts with 10Br<sup>-</sup> and 16H<sup>+</sup> to produce 2Mn<sup>2+</sup> and 5Br<sub>2</sub> with 8H<sub>2</sub>O as a byproduct.

MnO<sub>4</sub><sup>-</sup> first reacts with Fe<sup>2+</sup> in a 1:5 ratio, making the first equivalence point at 10 mL.

Once Fe<sup>2+</sup> is exhausted, MnO<sub>4</sub><sup>-</sup> reacts with Br<sup>-</sup> in a 2:10 ratio, adding another 25 mL for a total second equivalence point at 35 mL.

---

#### Few-shot 3 Query:

Consider a quantum mechanical system containing a particle of mass  $m$  moving in an isotropic three dimensional potential of the form  $V(r) = 1/2m\omega^2r^2$  corresponding to the acted force obeying Hooke's law. Here,  $\omega$  is the angular frequency of oscillation and  $r$  is the radial distance of the particle from the origin in spherical polar coordinate. What is the value of energy of the third excited state, and how many linearly independent eigenfunctions are possible for the same energy eigenvalue?

#### Few-shot 3 Response:

This problem is nothing but the three dimensional simple harmonic oscillator (SHO) problem.

The energy spectrum of three dimensional SHO is  $E_n = (n+3/2)\hbar\omega$  where  $n=0,1,2,3,\dots$ .

For third excited state  $n=3$ .

$3+3/2=6/2+3/2=9/2$ .

Thus the corresponding energy is  $(9/2)\hbar\omega$ .

The degeneracy of the state is  $g_n = (n+1)(n+2)/2$ .

For  $n=3$ , degeneracy is  $(3+1)*(3+2)/2=4*5/2=10$ .

---

#### Few-shot 4 Query:

Your overheard two chemists talking to each other as they leave a synthetic organic chemistry lab. One asks the other "So, how did it go?" The second chemist replies, "Not well - my compounds are on top of each other." What is the second chemist most likely referring to?

#### Few-shot 4 Response:

"On top of each other" commonly refers to two compounds that have similar R<sub>f</sub> values on chromatography (a common operation in synthetic chemistry).

Similar R<sub>f</sub> values arise for compounds with similar polarities.

---

#### Few-shot 5 Query:

Mitochondria are semi-autonomous cellular organelles in charge of energy production. They encode for a part of their own translational machinery and respiratory complexes. Mitochondrial function is governed by over a thousand proteins imported from the cell, contributing to processes like the transport of proteins, ribosome biogenesis and translation regulation, respiratory oxidation, metabolism, and apoptotic signaling cascade. Mutations in the code for mitochondrial protein networks can cause numerous diseases in humans that are inherited through generations. Mutations of which of the mitochondrial proteins listed below are least likely to be genetically transmitted from a father to his children?

#### Few-shot 5 Response:

The colleague should know that mitochondria from fathers are rarely if ever, transmitted to their offspring.

Therefore, the protein encoded by the paternal mitochondrial genome will most likely not be passed down the generation.

NADH dehydrogenase 2 is the only one encoded by the mitochondrial genome from the MT-ND2 gene among the listed proteins.

Leigh's syndrome, lactic acidosis, and metabolic diseases are all linked to a mutation in the ND2 gene.

ATP binding cassette subfamily B member 8 (ABCB8) is a chromosome 7 encoded gene; Tu translation elongation factor, mitochondrial is chromosome 16 gene TUFM.

Translocase of inner mitochondrial membrane 17B is chromosome X coded gene TIMM17B.

There is no evidence that it is maternally imprinted; hence, daughters may inherit the father's gene copy in a 50:50 ratio.

---

Table 32: Token-level qualitative analysis on HotpotQA.

---

Few-shot examples for HotpotQA (0.7883):

---

**Test-Query:**

H-63 is a highway that runs near the county seat of what Michigan county?.

---

**Few-shot 1 Query:**

What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

**Few-shot 1 Response:**

's think step by step. The eastern sector of Colorado orogeny extends into the High Plains. High Plains rise in elevation from around 1,800 to 7,000 ft, so the answer is 1,800 to 7,000 ft. Answer 1,800 to 7,000 ft

---

**Few-shot 2 Query:**

Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

**Few-shot 2 Response:**

's think step by step. Milhouse was named after U.S. president Richard Nixon, so the answer is Richard Nixon. Answer Richard Nixon

---

**Few-shot 3 Query:**

Which documentary is about Finnish rock groups, Adam Clayton Powell or The Saimaa Gesture?

**Few-shot 3 Response:**

's think step by step. Adam Clayton Powell (film) is a documentary about an African-American politician, not Finnish rock groups. So the documentary about Finnish rock groups must instead be The Saimaa Gesture. Answer The Saimaa Gesture

---

**Few-shot 4 Query:**

What profession does Nicholas Ray and Elia Kazan have in common?

**Few-shot 4 Response:**

's think step by step. Professions of Nicholas Ray are director, screenwriter, and actor. Professions of Elia Kazan are director, producer, screenwriter, and actor. So profession Nicholas Ray and Elia Kazan have in common is director, screenwriter, and actor. Answer director, screenwriter, actor

---

**Few-shot 5 Query:**

Which magazine was started first Arthur's Magazine or First for Women?

**Few-shot 5 Response:**

's think step by step. Arthur's Magazine was started in 1844. First for Women was started in 1989. 1844 (Arthur's Magazine) < 1989 (First for Women), so Arthur's Magazine was started first. Answer Arthur's Magazine

---

**Few-shot 6 Query:**

Were Pavel Urysohn and Leonid Levin known for the same type of work?

**Few-shot 6 Response:**

's think step by step. Pavel Urysohn is a mathematician. Leonid Levin is a mathematician and computer scientist. So Pavel Urysohn and Leonid Levin have the same type of work. Answer Yes

---

Table 33: Response-level qualitative analysis on MATH500.

## Responses for MATH500:

**Test-Query:**

Twelve 1 by 1 squares form a rectangle, as shown. What is the total area of the shaded region? [asy]

```
size(4cm);defaultpen(linewidth(0.75));

// Filled portions
fill((0,4)–(0,0)–(2,0)–cycle,gray(0.75));
fill((0,4)–(3,4)–(3,0)–cycle,gray(0.75));

// grid
int j;
for (j = 0; j < 4; ++j) draw((j,0)–(j,4));
for (j = 0; j < 5; ++j) draw((0,j)–(3,j));

//diagonals
draw((0,4)–(3,0)); draw((0,4)–(2,0));
[/asy]
```

**Worst Response (-1.736):**

To find the total area of the shaded region defined by the triangle at the top left and the rectangle on the right, we proceed as follows:

1. Identify the dimensions of the figure: The outline shows a rectangle formed by combining the squares. The rectangle consists of 3 squares in width (3 squares wide) and 4 squares in height (4 squares tall). Therefore, the rectangle has a total area of  $3 \times 4 = 12$  square units.

2. Determine the area of the upper triangle: The shaded area in the upper left can be identified as a right triangle, located at the top left corner of the rectangle. The base of this triangle is the height of the rectangle (4 squares), and the height is the width of the rectangle (2 squares). The area of a triangle is computed using the formula:

$$\text{Area of triangle} = \frac{1}{2} \times \text{base} \times \text{height}$$

For this triangle:

$$\text{Area} = \frac{1}{2} \times 2 \times 4 = \frac{1}{2} \times 8 = 4 \text{ square units}$$

3. Determine the area of the other shaded region: The other shaded area is the rectangle that extends across the top down to the base of the triangle. The rectangle also covers one of the 1x1 squares fully, specifically, it will cover the upper two squares fully which are not part of the triangle. This gives us a square area of  $3 \text{ (width)} \times 4 \text{ (height)} - \text{Area of triangle} = 12 - 4$ . Thus, the shaded area for consideration is:

$$12 - 4 = 8 \text{ square units}$$

4. Calculate the total shaded area: Adding these two areas we've established gives: - Area of upper triangle: 4 - Area of square: 8

However, we must ensure we are not double counting any area. Since the rectangle is inclusive, we need only evaluate the total shaded area: The shaded area includes just the triangle area since it encompasses all squares minus the interfered structured area.

The area accounted for based on shapes left is 4 described above in figures and does not undermine new counted elements.

5. Final answer: Thus, the area of the shaded region is confirmed as:

4

---

## Responses for MATH500:

---

### Best Response (-0.6992):

To find the total area of the shaded region in the rectangle formed by twelve 1 by 1 squares, let's analyze the problem step by step.

1. Understanding the Shape: The given rectangle consists of three columns and four rows of squares, leading to a total of  $3 \times 4 = 12$  unit squares.

2. Identifying the Shaded Areas: The shaded areas consist of two triangular regions: - The first triangular area is defined by points (0, 4), (0, 0), and (2, 0). - The second triangular area is defined by points (0, 4), (3, 4), and (3, 0).

3. Calculating the Area of the First Triangle: - Base = 2 (from (0, 0) to (2, 0)) - Height = 4 (from (0, 0) to (0, 4)) - Area of the triangle =  $\frac{1}{2} \times \text{Base} \times \text{Height} = \frac{1}{2} \times 2 \times 4 = 4$ .

4. Calculating the Area of the Second Triangle: - Base = 3 (from (3, 0) to (3, 4)) - Height = 4 (from (0, 4) to (0, 4)) - Area of the triangle =  $\frac{1}{2} \times \text{Base} \times \text{Height} = \frac{1}{2} \times 3 \times 4 = 6$ .

5. Adding the Areas: - Total shaded area = Area of first triangle + Area of second triangle =  $4 + 6 = 10$ .

Thus, the total area of the shaded region is 10.

---

Table 34: Response-level qualitative analysis on GPQA.

---

## Responses for GPQA:

---

### Test-Query:

Question: You have prepared an unknown compound. To identify the product, you have used the following characterisation techniques: FTIR and  $^1\text{H}$  NMR. The FTIR spectrum shows a very broad absorption peak at 3000 wavenumbers. A strong absorption peak is also observed at 1700 wavenumbers. Several peaks were observed in the  $^1\text{H}$  NMR spectrum, none of which are correspond to vinyl-hydrogens. One of the signals in the  $^1\text{H}$  NMR is a doublet of triplets of quartets whilst a different signal is a doublet of triplets of triplets. Identify the compound as either,  $\text{CH}_3\text{CH}_2\text{C}(\text{H})(\text{CH}_3)\text{C}(\text{H})(\text{CH}_3)\text{COOH}$ ,  $\text{CH}_3\text{CH}_2\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{COOH}$ ,  $\text{CH}_3\text{C}(\text{H})(\text{CH}_3)\text{C}(\text{H})(\text{CH}_3)\text{CH}_2\text{COOH}$ , or  $\text{CH}_3\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{CH}_2\text{COOH}$

---

---

## Responses for GPQA:

---

### **Worst Response (-1.344):**

Let's analyze the information provided systematically.

#### 1. **FTIR Analysis:**

- A broad absorption peak at 3000 wavenumbers indicates the presence of -OH or -NH groups, which are characteristic of carboxylic acids due to the hydrogen bonding which generates a broad peak.
- A strong absorption peak at 1700 wavenumbers suggests the presence of a carbonyl group (C=O), also associated with the carboxylic acid functional group.

#### 2. **<sup>1</sup>H NMR Analysis:**

- The absence of vinyl hydrogen signals suggests that none of the compounds contain a double bond (i.e., C=C).
- The signal described as a "doublet of triplets of quartets" indicates complex splitting patterns, likely resulting from hydrogens on adjacent carbon atoms that are not equivalent due to the molecular structure.
- The presence of various integrations and patterns suggests alkyl groups and potentially multiple types of adjacent hydrogens.

#### 3. **Evaluating the Choices:**

- Each choice represents a unique structure. We need to figure out which matches the spectral data:
- (A) CH3C(H)(CH3)C(H)(CH3)CH2COOH has the structure for 5 unique hydrogen environments and matches possible splitting patterns due to the complexity of the methyl and methylene groups.
- (B) CH3CH2C(H)(CH3)C(H)(CH3)COOH also has multiple potential splitting but is slightly less complex than (A).
- (C) CH3C(H)(C2H5)C(H)(C2H5)CH2COOH has an ethyl group that could create distinctive NMR peaks but still maintains hydrogen integration complexity.
- (D) CH3CH2C(H)(C2H5)C(H)(C2H5)COOH similarly maintains complexity, but having two ethyl groups may lead to a different integration pattern.

Given this information, we can further deduce that since the FTIR indicated a carboxylic acid and the NMR shows no vinyl protons, we can confirm the presence of -COOH in any of the compounds.

The heavy branching in (A) is consistent with the splitting observed (doublet of triplets of quartets) due to more complex interactions among various hydrogen types.

Considering the complexity and integration in the <sup>1</sup>H NMR, we can conclude that the most likely compound based on the spectral data is (A) because of the correct splitting patterns associated with the branching substitutions.

The correct answer is (A) CH3C(H)(CH3)C(H)(CH3)CH2COOH.

---

---

## Responses for GPQA:

---

### Best Response (-1.134):

To identify the unknown compound based on the provided spectroscopic data, let's analyze the information step by step.

#### 1. \*\*FTIR Analysis\*\*:

- A broad absorption peak at \*\*3000 wavenumbers\*\* indicates the presence of an -OH group, typical of carboxylic acids or alcohols.
- A strong absorption peak at \*\*1700 wavenumbers\*\* suggests a carbonyl (C=O) group, common in carboxylic acids and aldehydes.

#### 2. \*\*1H NMR Analysis\*\*:

- The absence of vinyl hydrogens indicates there are no double bonds present in the compound.
- The \*\*doublet of triplets of quartets\*\* and a \*\*doublet of triplets of triplets\*\* indicate a relatively complex splitting pattern which typically arises from protons in different environments due to adjacent protons.

#### 3. \*\*Analysis of the compounds\*\*:

- The compounds listed can be broken down as follows:
  - (A)  $\text{CH}_3\text{C}(\text{H})(\text{CH}_3)\text{C}(\text{H})(\text{CH}_3)\text{CH}_2\text{COOH}$ : A branched structure likely leading to complex splitting.
  - (B)  $\text{CH}_3\text{CH}_2\text{C}(\text{H})(\text{CH}_3)\text{C}(\text{H})(\text{CH}_3)\text{COOH}$ : A linear structure with groups that may contribute to the complexity in splitting.
  - (C)  $\text{CH}_3\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{CH}_2\text{COOH}$ : Also branched, with ethyl groups that could lead to intricate splitting.
  - (D)  $\text{CH}_3\text{CH}_2\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{C}(\text{H})(\text{C}_2\text{H}_5)\text{COOH}$ : A linear structure including ethyl groups, likely resulting in simpler splitting.

#### 4. \*\*Choosing the Compound\*\*:

- The broad absorption at 3000  $\text{cm}^{-1}$  and significant peaks at 1700  $\text{cm}^{-1}$  underscores that we are likely dealing with a carboxylic acid, given it shows a carbonyl stretch and alcohol absorbance.
- The complexity of the NMR signals suggests a branched structure as those are usually more complex due to the adjacent hydrogen interactions.

#### 5. \*\*Final Decision\*\*:

- \*\*Compound (C)\*\*, containing ethyl groups and overall complex splitting, aligns with the described spectroscopic data. The branching and additional substituents in compound (C) can contribute to complex resonance, further supported by the FTIR properties.

Thus, based on the analysis of FTIR and NMR data, the compound can be most accurately identified as:

The correct answer is (C).

---

Table 35: Response-level qualitative analysis on HotpotQA.

---

## Responses for HotpotQA:

---

### Test-Query:

H-63 is a highway that runs near the county seat of what Michigan county?

---

### Worst Response (-1.324):

Let's think step by step. The H-63 highway runs in Michigan near the county seat of Baraga County, which is L'Anse. So, H-63 is associated with Baraga County.

Answer Baraga County

---

### Best Response (-1.033):

Let's think step by step. H-63 is a highway located in Michigan. To determine the county seat it runs near, we can look at its route. H-63 runs primarily through Chippewa County in the Upper Peninsula of Michigan. The county seat of Chippewa County is Sault Ste. Marie. Therefore, H-63 is a highway that runs near the county seat of Chippewa County.

Answer Chippewa County

---

---

**Listing 1** Few-shot CoT prompt on MATH500

---

```
f'''
{system}
Please reason step by step, and put your final answer within \boxed{{}}.
-----
{user}
Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only
↪  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$ 
↪ of the remaining distance. How far has he hopped after five hops? Express your answer as a
↪ common fraction.

Let's think step by step
Kevin hops  $\frac{1}{3}$  of the remaining distance with every hop.
His first hop takes  $\frac{1}{3}$  closer.
For his second hop, he has  $\frac{2}{3}$  left to travel, so he hops forward  $(\frac{2}{3})(\frac{1}{3})$ .
For his third hop, he has  $(\frac{2}{3})^2$  left to travel, so he hops forward  $(\frac{2}{3})^2(\frac{1}{3})$ .
In general, Kevin hops forward  $(\frac{2}{3})^{k-1}(\frac{1}{3})$  on his  $k$ th hop.
We want to find how far he has hopped after five hops.
This is a finite geometric series with first term  $\frac{1}{3}$ , common ratio  $\frac{2}{3}$ , and five terms.
Thus, Kevin has hopped  $\frac{1}{3}\left(1-\left(\frac{2}{3}\right)^5\right)$ 
 $1-\frac{2}{3} = \boxed{\frac{1}{3}}$ .
The answer is  $\frac{1}{3}$ 

...

Convert the point  $(0,3)$  in rectangular coordinates to polar coordinates. Enter your answer in the
↪ form  $(r,\theta)$ , where  $r > 0$  and  $0 \leq \theta < 2\pi$ .
'''
```

---

---

**Listing 2** Zero-shot CoT prompt on MATH500

---

```
f'''
{system}
Please reason step by step, and put your final answer within \boxed{{}}.
-----
{user}
Convert the point  $(0,3)$  in rectangular coordinates to polar coordinates. Enter your answer in the
↪ form  $(r,\theta)$ , where  $r > 0$  and  $0 \leq \theta < 2\pi$ .
'''
```

---

---

**Listing 3** Prompt for USC

---

```
f'''
I have generated the following responses to the question: Convert the point  $(0,3)$  in rectangular
↪ coordinates to polar coordinates. Enter your answer in the form  $(r,\theta)$ , where  $r > 0$  and
↪  $0 \leq \theta < 2\pi$ .

Response 0: {response0}

...

Response 4: {response4}

Evaluate these responses.
Select the most consistent response based on majority consensus.
Start your answer with "The most consistent response is Response X" (without quotes).
'''
```

---

---

**Listing 4** Prompt for LEAP mistakes

---

```
f'''
{system}
Please reason step by step, and put your final answer within \boxed{{}}.
-----
{user}
Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only
↪  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$ 
↪ of the remaining distance. How far has he hopped after five hops? Express your answer as a
↪ common fraction.
'''
```

---

---

**Listing 5** Prompt for LEAP low-level principles

---

```
f'''
Question: {question}
Generated Reasoning: {response}

Generated Answer: {generated_answer}

Correct Reasoning: {correct_reasoning}

Correct Answer: {correct_answer}

Instruction: Conduct a thorough analysis of the generated answer in comparison to the correct answer.
↪ Also observe how the generated reasoning differs from the correct reasoning. Identify any
↪ discrepancies, misunderstandings, or errors. Provide clear insights, principles, or guidelines
↪ that can be derived from this analysis to improve future responses. We are not focused on this
↪ one data point, but rather on the general principle.

Reasoning: <discuss why the generated answer is wrong>
Insights: <what principle should be looked at carefully to improve the performance in the future>
'''
```

---

---

**Listing 6** Prompt for LEAP high-level principles

---

```
f'''
Low-level principles:
{low_level_principles}

Create a list of *unique* and insightful principles to improve future responses based on the
↪ analysis above.
Focus on capturing the essence of the feedback while eliminating redundancies.
Ensure that each point is clear, concise, and directly derived from the introspection results.
Create a numbered list of principles. Leave specific details in place.
Limit to at most 8 principles.

List of Principles:
'''
```

---

---

**Listing 7** Prompt for LEAP generations

---

```
f'''
{system}
Please reason step by step, and put your final answer within \boxed{{}}.
-----
{user}
Please carefully note the following principles:

Principles: 1. Meticulous Verification: Always verify each step in algebraic processes to
↳ prevent errors that can lead to incorrect conclusions.

...

8. Continuous Learning and Adaptation: Stay open to learning from mistakes and adapting methods
↳ to improve future problem-solving approaches.

Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only
↳  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$ 
↳ of the remaining distance. How far has he hopped after five hops? Express your answer as a
↳ common fraction.

Let's think step by step
Kevin hops  $\frac{1}{3}$  of the remaining distance with every hop.
His first hop takes  $\frac{1}{3}$  closer.
...

Convert the point  $(0,3)$  in rectangular coordinates to polar coordinates. Enter your answer in the
↳ form  $(r,\theta)$ , where  $r > 0$  and  $0 \leq \theta < 2\pi$ .
'''
```

---

---

**Listing 8** Prompt for USC-w/ Fewshot

---

```
f'''
Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only
↳  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$ 
↳ of the remaining distance. How far has he hopped after five hops? Express your answer as a
↳ common fraction.

Let's think step by step
Kevin hops  $\frac{1}{3}$  of the remaining distance with every hop.
His first hop takes  $\frac{1}{3}$  closer.
...

I have generated the following responses to the question: Convert the point  $(0,3)$  in rectangular
↳ coordinates to polar coordinates. Enter your answer in the form  $(r,\theta)$ , where  $r > 0$  and
↳  $0 \leq \theta < 2\pi$ .

Response 0: {response0}

...

Response 4: {response4}

Evaluate these responses.
Select the most consistent response based on majority consensus.
Start your answer with "The most consistent response is Response X" (without quotes).
'''
```

---

---

**Listing 9** Prompt for LLM-as-Judge

---

```
f'''
{system}
Your job is selecting the most accurate response among multiple candidates. You will receive a
↪ question and several candidate answers labeled candidate1, candidate2, etc. Please summarize the
↪ debate very briefly and then conclude which single candidate is the most plausible. Output
↪ exactly in this format:
Summary: <brief summary>
Conclusion: candidate<number>
Remember to choose only one candidate as the final answer.
-----

{user}
Please reason step by step, and put your final answer within \boxed{{}}.

The below examples are well-constructed gold question and answer pairs for the same task.

Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only
↪  $\frac{1}{3}$  of the distance. Each hop tires him out so that he continues to hop  $\frac{1}{3}$ 
↪ of the remaining distance. How far has he hopped after five hops? Express your answer as a
↪ common fraction.

Let's think step by step
Kevin hops  $\frac{1}{3}$  of the remaining distance with every hop.
His first hop takes  $\frac{1}{3}$  closer.
...

Now, let's select the most proper answer for the given question
Question: Convert the point  $(0,3)$  in rectangular coordinates to polar coordinates. Enter your
↪ answer in the form  $(r,\theta)$ , where  $r > 0$  and  $0 \leq \theta < 2\pi$ .
candidate1: {response 0}
...
candidate5: {response 4}
'''
```

---

---

**Listing 10** Prompt for generate weak few-shot

---

```
f'''
"You will receive a QUESTION and its original ANSWER.\n"
"Rewrite ONLY the ANSWER; do NOT alter the QUESTION.\n"
"Treat the original as a 10/10 reference. Produce a deliberately degraded explanation (target
↪ quality 1/10):\n"
"- Keep the final answer tokens EXACT (e.g., '\boxed{...}' or 'The correct answer is (X)').\n"
"- Keep the original CoT style label if present (e.g., 'Let's think step by step:' /
↪ 'Reasoning:').\n"
"- Make reasoning weak: shallow, vague, incomplete; omit steps, avoid precise formulas/numbers.\n"
"- Prefer generic phrases over concrete derivations. Lower clarity and rigor compared to the
↪ original.\n"
"OUTPUT FORMAT: Return EXACTLY ONE JSON object and NOTHING ELSE:\n"
'{"answer": "<rewritten weaker answer>"}'
'''
```

---

---

**Listing 11** Prompt for judging weak few-shot

---

```
f'''
    "You are judging FEW-SHOT QUALITY only.\n"
    "Compare TWO blocks side-by-side. Assume BLOCK A and BLOCK B are candidate few-shot
    ↪ demonstrations.\n\n"
    "Ignore question quality entirely – the question is context only.\n\n"

    "What “answer quality” means here:\n"
    "- clarity, structure, and coherence of the reasoning.\n"
    "- specific steps, concrete numbers/equations when relevant, and justified transitions.\n"
    "- a single, clearly marked final answer token format (e.g., \"\boxed{...}\") or \"The correct
    ↪ answer is (X)\" if present;\n"

    "Instructions:\n"
    "- Assign an integer score 1-10 to EACH block (higher = better few-shot quality).\n"
    "- The evaluation should be comparative: scores must reflect their relative quality.\n"
    "- Provide brief notes explaining each score.\n\n"
    "OUTPUT FORMAT:\n"
    "Return exactly ONE JSON object with this schema (and nothing else):\n"
    "{"
    "\"A\":{\"score\":int,\"notes\":string},"
    "\"B\":{\"score\":int,\"notes\":string},"
    "\"comparative_notes\":string"
    "'''
```

---

---

**Listing 12** Prompt for plan-and-solve on MATH500

---

```
f'''
    "Let's first understand the problem, extract relevant variables and their corresponding numerals,
    ↪ and make a complete plan. Then, let's carry out the plan, calculate intermediate variables (pay
    ↪ attention to correct numerical calculation and commonsense), solve the problem step by step, and
    ↪ put your final answer within \boxed{{}}.\n"
    "'''
```

---

---

**Listing 13** Prompt for role-playing on MATH500

---

```
f'''
    "From now on, you are an excellent math teacher and always teach your students math problems
    ↪ correctly. And I am one of your students. Put your final answer within \boxed{{}}.\n"
    "'''
```

---