
Training-free LLM Verification via Recycling Few-shot Examples

Dongseok Lee^{1,2} Jimyung Hong² Dongyoung Kim³ Jaehyung Kim²

Abstract

Although LLMs have achieved remarkable performance, the inherent stochasticity of their reasoning process and varying conclusions present significant challenges. Majority voting or Best-of-N with external verification models has been explored to find the most promising solution among multiple LLM outputs. However, these approaches have certain limitations, such as limited applicability or the cost of an additional training step. To address this problem, we propose a novel and effective framework that **Recycles Few-shot examples to verify LLM outputs (ReFeri)**. Our key idea is to additionally utilize the given few-shot examples to evaluate the candidate outputs of the target query, not only using them to generate outputs as the conventional few-shot prompting setup. Specifically, ReFeri evaluates the generated outputs by combining two different scores, designed motivated from Bayes' rule, and selects the candidate that is both confidently determined and contextually coherent through a few additional LLM inferences. Experiments with three different LLMs and across seven diverse tasks demonstrate that our framework significantly improves the accuracy of LLMs-achieving an average gain of 4.8%-through effective response selection, without additional training.

1. Introduction

Recently, large language models (LLMs) have shown remarkable performance in many real-world tasks involving complex reasoning such as math, coding, and robotics (Anthropic, 2024; Dubey et al., 2024; OpenAI, 2024c; Team et al., 2023). To enhance the reasoning capacity of LLMs,

various approaches have been proposed from in-context learning at test time (Wei et al., 2022; Kojima et al., 2022) to recent RL training method (Qu et al., 2024; Guo et al., 2025). Despite these improvements, the inherent stochastic nature of LLM still presents significant challenges, since different reasoning paths can be generated for the same input and can lead to varying conclusions (Kadavath et al., 2022; Wang & Zhou, 2024; Qiu & Miiikkulainen, 2024). Majority voting approaches, such as self-consistency (Wang et al., 2023; Aggarwal et al., 2023), have been widely adopted to reduce such randomness by aggregating multiple LLM outputs and determining a single prediction. However, this approach is only applicable when the answer can be easily extracted from the output and aggregated. Consequently, it is difficult to apply to open-ended text generation tasks such as summarization and personalized chatbot (Stiennon et al., 2020; Salemi et al., 2024).

To address this challenge, finding the most promising one among multiple LLM outputs using a specific selection method, often called *Best-of-N*, has recently gained attention (Snell et al., 2024; Gui et al., 2024). For instance, one of the most representative approaches is to score each output using external verification models such as Outcome Reward Models (ORMs) (Cobbe et al., 2021; Uesato et al., 2022) or Process Reward Models (PRMs) (Lightman et al., 2024; Wang et al., 2024a), and then selecting the highest-scoring output. However, to obtain these reward models, training with a large amount of task-specific labeled data is often necessary; therefore, applying this framework to specific target domain, which is far from well-explored domains such as math and coding, is challenging. Prompting LLM to select the most promising output such as *LLM-as-judge* is considerable to remove the reliance on the verification model (Chen et al., 2023; Zheng et al., 2023). However, this approach is only effective when the given LLM has sufficient intrinsic knowledge for the target domain; consequently, it often requires separate training steps and datasets again to achieve sufficient performance (Yuan et al., 2024; Mahan et al., 2024; Zhang et al., 2025).

Motivated by this, we suggest a new perspective: *utilization of few-shot examples to verify and select among multiple LLM outputs*. As recent LLMs have been trained with an

¹This work was done while the author was an intern at Yonsei ML3. ²Yonsei University ³KAIST. Correspondence to: Jaehyung Kim <jaehyungk@yonsei.ac.kr>.

extensive instruction tuning step, they often exhibit better performance without few-shot examples (Guo et al., 2025; Sprague et al., 2025), and hence using these examples at test time is recently losing attention (see results in Table 1). However, we argue that using few-shot examples is still one of the easiest and most direct ways to let LLMs know how to solve the given task with human prior knowledge, even if LLMs have not encountered it before. Therefore, in this work, we provide a new framework that enables better exploitation of few-shot examples by using them not only for generating multiple outputs, but also for selecting the most promising one.

Contribution. In this work, we propose **ReFeri**, a novel and effective framework that **Recycles Few-shot** examples to **verify** LLM outputs. The core idea of ReFeri is additionally utilizing the given few-shot examples to evaluate the candidate outputs of the target query, not only using them to generate outputs as conventional few-shot in-context learning.¹ Specifically, ReFeri estimates the likelihood of the generated outputs by decomposing it into two different scores conditioned on few-shot examples, which are derived from *Bayes’ rule*. The forward confidence score measures the likelihood of candidate outputs given the few-shot examples and the test query, favoring more confident ones. On the other hand, the backward consistency score measures whether conditioning on the candidate output affects the likelihood of the few-shot examples compared to conditioning on their queries alone. By combining these scores, ReFeri selects the candidate that is both confidently determined and contextually coherent through a few additional LLM inferences. Consequently, ReFeri does not require additional model training to select the most promising output, and allows better leverage of both intrinsic knowledge of LLM and human prior within the provided few-shot examples. See Figure 1 for the illustration.

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 (CoT) prompting, ReFeri consistently outperforms other training-free selection across all tasks, with an average gain of 4.8% over

random selection and 2.6% over prompt-based selection methods (see Figure 2). ReFeri also scales reliably with the number of candidate responses, demonstrating its practical utility in test-time scaling. To better understand the behavior of ReFeri, we conduct two complementary analyses, showing that our method is robust to both variations in few-shot example selection and the choice of model used for likelihood estimation. Furthermore, we evaluate ReFeri in more challenging and practical scenarios. Although recent reasoning LLMs (Guo et al., 2025) show degraded performance under a few-shot prompting, ReFeri consistently finds out the most promising reasoning path. Moreover, ReFeri shows effectiveness on LLM personalization (Salemi et al., 2024) requires open-ended, user-specific generation.

2. Related Works

Few-shot in-context learning of LLM. Few-shot in-context learning (ICL) revealed that LLMs can generalize to unseen tasks with just a handful of input-output demonstrations (Brown et al., 2020). To handle complex reasoning problems, chain-of-thought (CoT) prompting was proposed to append intermediate steps to the few-shot examples, leading to substantial gains in tasks such as arithmetic, common-sense reasoning, and symbolic manipulation (Wei et al., 2022; Fu et al., 2023; Jin et al., 2024). To further enhance ICL, various strategies have been developed to retrieve better examples using semantic similarity or entropy-based selection (Wu et al., 2023; Peng et al., 2024). However, some studies have shown that few-shot ICL does not always guarantee improvements. For instance, label shuffling or format changes can often leave performance unaffected (Min et al., 2022), and the performance gap between zero-shot and few-shot CoT is narrowing in several benchmarks as instruction tuning becomes more effective (Sprague et al., 2025). In particular, recent LLMs such as DeepSeek-R1, which are trained with reinforcement learning-based reasoning steps, sometimes even show performance degradation when few-shot CoT examples are added (Guo et al., 2025). Nonetheless, carefully selected demonstrations are still effective (Huang et al., 2024). For example, (Ge et al., 2025) show that few-shot examples can reduce overconfidence in multi-step reasoning, and (Yan et al., 2025) show that they help mitigate hallucinations and memory-based mistakes in complex tasks. These observations motivate us to go beyond using few-shot examples for generation, and recycling few-shot examples to evaluate the possible multiple LLM responses and to select the most promising one.

Selection of diverse LLM outputs. Due to the probabilistic nature of LLM decoding, LLM can provide diverse outputs for a single input, each reflecting different reasoning paths (Kadavath et al., 2022; Wang & Zhou, 2024; Qiu

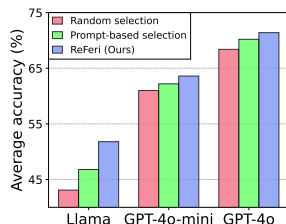


Figure 2. Summary of results. Average accuracy across seven benchmarks with training-free selection methods. ReFeri shows consistent effectiveness (see Section 4.2).

¹In-context learning uses given few-shot examples as additional input context upon the target query.

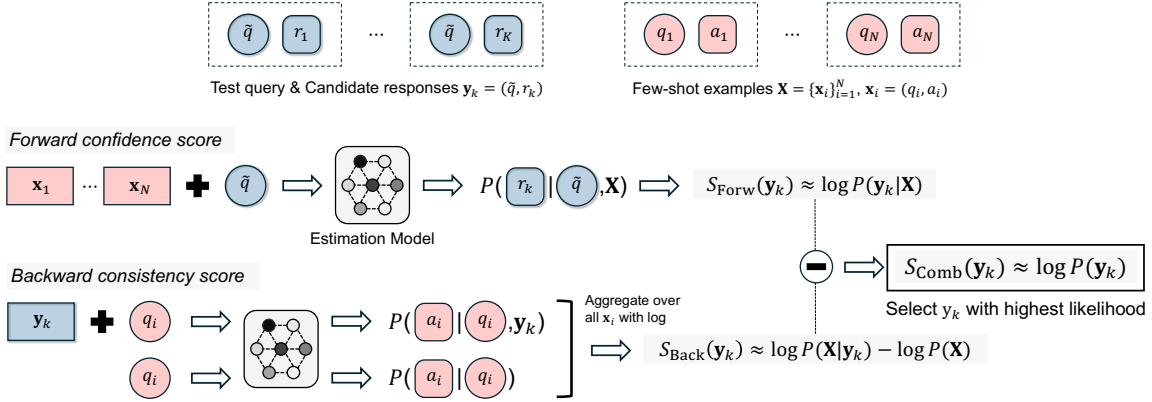


Figure 1. **An overview of ReFeri.** For K candidate responses from LLMs, ReFeri assigns each candidate a forward confidence score (how likely candidate is to be generated conditioned on few-shot examples) and a backward consistency score (how candidate affect to reconstruct the answers of few-shot examples). Then, the response with the best joint score is selected as the final answer.

& Miikkulainen, 2024). To handle this variability, self-consistency (Wang et al., 2023) samples K independent reasoning paths and selects the majority answer to improve accuracy. However, it assumes that the model produces a single, well-formatted answer, and this assumption is often violated in open-ended tasks such as summarization or free-form dialogue (Stiennon et al., 2020; Salemi et al., 2024). Alternatively, recent Best-of-N approaches aim to directly select the best output among candidates, often using external verification models. For instance, Outcome Reward Models (ORMs) grade final outputs (Cobbe et al., 2021; Uesato et al., 2022), while Process Reward Models (PRMs) assess intermediate reasoning steps to provide finer supervision (Lightman et al., 2024; Wang et al., 2024a). Despite their successes, these models require large-scale, task-specific annotations or domain-specific checkers, limiting their scalability to new domains or unseen tasks. To eliminate the need for external verification models, prompting-based methods such as LLM-as-Judge ask LLM to evaluate its own outputs (Chen et al., 2023; Zheng et al., 2023). However, their effectiveness heavily depends on the model’s prior knowledge in the target domain. When this knowledge is lacking, these methods require additional fine-tuning with curated evaluation datasets for sufficient performance, which reintroduces the need for supervision (Yuan et al., 2024; Mahan et al., 2024; Zhang et al., 2025). In contrast, ReFeri is training-free and task-agnostic by recycling a few-shot examples for verification.

3. Training-free Verification of LLM Outputs via Recycling Few-shot Data

3.1. Preliminary

Let us denote LLM as \mathcal{M} and a given test query as \tilde{q} . We assume that we have N -shot examples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i = (q_i, a_i)$ where q_i is another input query from the same task

and a_i is the ground-truth answer, which can be provided by human annotator or generated by LLM itself. Then, *few-shot prompting* incorporates the few-shot examples \mathbf{x}_i in \mathbf{X} as additional input context to obtain the response r_k , which is expected to be improved thanks to the in-context learning capability of LLMs:

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

where multiple non-identical predictions $r_k, k = 1, \dots, K$ can be sampled. Then, our goal is to find the most appropriate response r_{k^*} among them. For example, the self-consistency method (Wang et al., 2023) simply applies majority voting to determine the single prediction. On the other hand, the best-of-K method uses the external verifier such as reward models (Cobbe et al., 2021; Lightman et al., 2024) to score the predictions and select the highest scored one. Formally, with the external verifier R_ϕ , it can be described as below:

$$r_{k^*} = \arg \max_{k=1, \dots, K} R_\phi(\mathbf{y}_k), \quad (2)$$

where $\mathbf{y}_k = (\tilde{q}, r_k)$. While these approaches are widely used in practice, there are certain challenges due to the limited applicability and the need for a verification model for the target task.

3.2. ReFeri: Verification of LLM outputs with Bayes-inspired scores with few-shot data

In this section, we introduce a framework that selects candidates from LLM by **Recycling Few-shot examples** for the verification (**ReFeri**). The core idea of ReFeri is to leverage few-shot examples not only for generation but also for validation, thereby recycling them to score and select answers without additional training. Specifically, ReFeri estimates the plausibility of each answer candidate by combining two complementary signals: (1) a *forward confidence score* which captures how likely the model is to generate response r_k given test query \tilde{q} , few-shot examples \mathbf{X} , and (2)

a *backward consistency score*, measuring how r_k is effective to correctly answer the queries q_i in \mathbf{X} .

Problem setup. Let us assume that we have an estimation model P which can measure the likelihood $P(\mathbf{y}_k) = P(r_k | \tilde{q})$ of the response r_k conditioned on the given query \tilde{q} .² Then, our goal is to select the response r_{k^*} which yields the highest likelihood if the estimation is accurate:

$$k^* = \arg \max_{k=1, \dots, K} P(\mathbf{y}_k). \quad (3)$$

We note that the likelihood has shown effectiveness to find high-quality reasoning path (Wang & Zhou, 2024). However, selecting based on the estimated $P(\mathbf{y}_k)$ could be ineffective in practice, as it entirely depends on the estimation model’s intrinsic knowledge, which can be limited in unfamiliar or challenging domains. Furthermore, when there is a mismatch between \mathcal{M} and P , the estimated likelihoods can be unreliable as minor syntactic variations in response can make large deviations. To address this, we propose to reinterpret $P(\mathbf{y}_k)$ with few-shot examples \mathbf{X} , through Bayes’ rule:

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

Then, in the log form, this can be decomposed into two intuitive forward and backward scores:

$$\begin{aligned} \log P(\mathbf{y}_k) &= \underbrace{\log P(\mathbf{y}_k | \mathbf{X})}_{\text{forward}} \\ &\quad - \underbrace{(\log P(\mathbf{X} | \mathbf{y}_k) - \log P(\mathbf{X}))}_{\text{backward}}. \end{aligned} \quad (5)$$

While Eq. 5 holds mathematically, discrepancies between the left- and right-hand sides can arise in practice due to the limitations of the estimation model. To address this, the core idea of ReFerri is to estimate the forward and backward scores separately, as each can be more accurately approximated by the estimation model with the help of few-shot examples. Then, ReFerri combines these two estimated scores to yield the final selection score. Overall algorithm is presented in Algorithm 1.

Forward confidence score. Intuitively, $\log P(\mathbf{y}_k | \mathbf{X})$ captures the confidence of candidate response r_k to test query \tilde{q} ; this score is high when r_k well-aligns with the reasoning patterns in the few-shot examples \mathbf{X} . This forward score has certain advantages over direct estimation of $P(\mathbf{y}_k)$, 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. When the estimation model P

²For the experiments in Section 4, we use pre-trained LLM as the estimation model.

is equal to generation LLM \mathcal{M} , the forward score can be freely obtained during generation of r_k . Formally, under the autoregressive assumption for estimation model P , the forward score is derived as below:

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

where each candidate response is a sequence of T tokens $r_k = (r_{k,1}, \dots, r_{k,T})$.

Backward consistency score. The backward score, $\log P(\mathbf{X} | \mathbf{y}_k) - \log P(\mathbf{X})$, evaluates how the inclusion of test query \tilde{q} and candidate response r_k affect to explain the few-shot examples \mathbf{X} . At a high level, this score serves as a form of consistency check between the response and the given few-shot examples. Under the assumption of mutual independence between few-shot examples, the backward score can also be derived similar to Eq. 6. However, to better evaluate how well the candidate response r_k explains the few-shot examples \mathbf{X} , we refine the backward term using a leave-one-out strategy (Perez et al., 2021; Izacard et al., 2023) through prompt replacement; namely, we construct new demonstration $\tilde{\mathbf{X}}_i$ by replacing i -th example $\mathbf{x}_i = (q_i, a_i)$ with a pair of test query and candidate response (\tilde{q}, r_k) :

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

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

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

This inclusion of remaining examples \mathbf{X}_{-i} enables more accurate estimation of the likelihood of target example \mathbf{x}_i by leveraging the in-context learning capability of P (see more discussions in Appendix B.2). Similar to Eq. 6, $\log P(a_i | q_i, \tilde{\mathbf{X}}_i)$ and $\log P(a_i | q_i)$ can be calculated through a token-level decomposition using the autoregressive nature of P .

Final score. By combining forward and backward scores following Eq. 5, we design our main selection score S_{Comb} to find the most promising output r_{k^*} as below:

$$\begin{aligned} k^* &= \arg \max_{k=1, \dots, K} S_{\text{Comb}}(\mathbf{y}_k), \\ S_{\text{Comb}}(\mathbf{y}_k) &:= S_{\text{Forw}}(\mathbf{y}_k) - S_{\text{Back}}(\mathbf{y}_k). \end{aligned} \quad (9)$$

However, the direct estimation of $P(\mathbf{y}_k)$ can be effective, particularly when both the estimation model P and the generation LLM \mathcal{M} perform reliably—as is often the case in

well-covered, generic domains. Therefore, to utilize this information upon the proposed combined score S_{Comb} , we define the final score S_{Fin} by jointly considering both components as follows:

$$S_{\text{Fin}}(\mathbf{y}_k) := (1 - \beta) \cdot S_{\text{Dire}}(\mathbf{y}_k) + \beta \cdot S_{\text{Comb}}(\mathbf{y}_k), \quad (10)$$

where $S_{\text{Dire}} = P(\mathbf{y}_k)$ and β is a hyperparameter to control the effect from two sources. This is helpful to prevent over-reliance on noisy few-shot examples of S_{Comb} and improves the robustness of selection. When $\beta = 1$, the score relies fully on the score derived forward-backward decomposition.

4. Experiments

In this section, we design our experiments to investigate the following questions:

- Is ReFeri effective to select the correct output across various tasks and LLMs? (Table 1)
- Can ReFeri enable test-time scaling without external reward model and training? (Figure 3)
- What is the effect of each component in ReFeri? (Tables 2, 3, and Figure 4)
- Can ReFeri be generalized to recent reasoning LLM and broader tasks? (Tables 4, 5)

4.1. Setups

Datasets. We evaluate our method on seven benchmarks encompassing diverse reasoning paradigms, including symbolic-numeric, expertise-based, and multi-hop textual reasoning tasks. (1) *MATH500* (Lightman et al., 2024); a 500-problem subset of the MATH benchmark (Hendrycks et al., 2021b), focused on symbolic manipulation and multi-step mathematical reasoning. (2) *MMLU-pro* (Wang et al., 2024b); 4200 examples, including 300 randomly sampled questions per domain (e.g., physics, law, computer science) extends the original MMLU benchmark (Hendrycks et al., 2021a) by adding reasoning-focused questions and expanding the choice set from four to ten options. (3) *HotpotQA* (Yang et al., 2018); 500 samples from (Kim et al., 2024) which is a multi-hop question-answering benchmark requiring reasoning across multiple Wikipedia paragraphs with annotated supporting facts. (4) *DROP* (Dua et al., 2019); 500 randomly sampled questions from this reading comprehension benchmark, which queries demand discrete numerical reasoning (e.g., addition, counting, sorting) over paragraphs. (5) *GPQA-diamond* (Rein et al., 2024) (*GPQA*); 198 graduate-level questions assessing complex reasoning in biology, physics, and chemistry. (6,7) *MuSR* (Sprague et al., 2024); 256 examples in Object Placement (*MuSR-op*) and 250 examples in Team Allocation (*MuSR-ta*) tasks assessing spatial and relational reasoning.

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. As few-shot examples are necessary for some baselines and ReFeri, we collect them following the previous works. *MATH500*: 5 examples from (Yang et al., 2024) (GPTs), 4 examples from (Lewkowycz et al., 2022) (LLaMA).³ *MMLU-Pro*: 5 examples from (Wang et al., 2024b). *HotpotQA*: 6 examples from (Yao et al., 2023). *DROP*: 3 examples following (Zhou et al., 2022). *GPQA-Diamond*: 5 examples from (Rein et al., 2024). *MuSR*: 3 examples from (Sprague et al., 2025). Complete prompt templates are available in Appendix A.1.

Baselines. We compare ReFeri against five prompt-based methods that require no additional training, with some reflecting different uses of few-shot examples: (1) *Zero-shot CoT* appends a trigger phrase (“Let’s think step by step.”) to each query without providing exemplars, relying on LLM’s intrinsic reasoning capabilities. (2) *Few-shot CoT* prepends a fixed set of few examples, enabling LLM to generalize from few in-context demonstrations. (3) *USC* asks LLM to select the best answer from multiple CoT outputs, by following (Chen et al., 2023). We report the results without using few-shot examples as it degrades performance (see Appendix B.3). (4) *CoT-WP* (Wang & Zhou, 2024) scores each candidate response using token-level probabilities from LLM conditioned on the same few-shot examples. Specifically, the score is a confidence gap between top-1 and top-2 tokens at answer positions. To estimate confidence, we use LLaMA-3.1-8B as same as ReFeri. (5) *LEAP* (Zhang et al., 2024) improves few-shot prompting by intentionally inducing mistakes on few examples. Then extracting generalizable task-specific principles through self-reflection without human annotations, and prompting the model to apply these principles to unseen questions. Specific prompts for each baseline are in Appendix A.2.

Implementation details. For the experiments, we use (1) gpt-4o-2024-08-06 (*GPT-4o*) (OpenAI, 2024a), (2) gpt-4o-mini-2024-07-18 (*GPT-4o-mini*) (OpenAI, 2024b), and (3) LLaMA-3.1-8B-Instruct (*LLaMA-3.1-8B*) (Dubey et al., 2024) as target LLMs, i.e., response generation models. We generate $K = 5$ responses per each query using temperature of 1.0 to encourage diverse candidates. For Zero-shot CoT, Few-shot CoT and LEAP, we report the average accuracy for all five responses without applying any selection mechanism, which can be viewed as randomly selecting the response. For USC, CoT-WP, and ReFeri, we use the same candidates generated from Few-shot CoT. In USC, the generation and judge models are same and we use temperature of 0. For the estimation model P , we employ LLaMA-3.1-8B-Instruct, except the experiments in Figure 4. For the hyper-parameter of ReFeri, we consider

³(1) Using the same prompt as GPT results in significantly lower accuracy, and (2) LLaMA-based models provide their own optimized prompt templates (see meta-llama/Llama-3.2-3B-Instruct-evals).

$\beta \in [0.5, 0.75, 1]$. More details are in Appendix A.3.

4.2. Main results

Table 1 summarizes the experimental results across seven different reasoning benchmarks and three different LLMs. For instance, across all tested LLMs and benchmarks, ReFeri improves average accuracy by 4.8% over Few-shot CoT, which corresponds to apply random selection instead. Compared to the second-best method, CoT-WP, ReFeri achieves an average improvement of 1.7% across all benchmarks. Notably, CoT-WP relies solely on the forward likelihood of each candidate, while ReFeri combines both forward and backward signals via a Bayes-derived scoring function. This bidirectional formulation allows ReFeri to capture not just the confidence of an answer, but also its consistency with few-shot examples upon the LLM’s intrinsic knowledge about the task; consequently, it enables a better selection across various tasks. We note that performance of prompt-base selection, USC, largely varies depending on the task and used LLMs, which reveals the limitation of solely relying on LLM’s intrinsic knowledge. In addition, as mentioned in Section 4.1, MuSR is a benchmark where few-shot examples play a critical role (Sprague et al., 2025) and our results also support this with 21.0% average improvement by Few-shot CoT over Zero-shot CoT. Here, we find that ReFeri further enlarges the improvement with the largest gain, outperforming the second-best method by 4.5%. This result shows that ReFeri is particularly effective in new domains where there LLM has a little prior knowledge and need to heavily rely on a few examples without additional training or reward models.

Next, to assess whether ReFeri scales effectively with the number of candidate outputs similar to the conventional reward-based best-of- K selection, we evaluate performance as the candidate pool grows. Specifically, we test $K = \{1, 5, 10, 15\}$ candidates on three representative tasks—*MATH500*, *GPQA*, and *MuSR-ta* by using GPT-4o-mini as the generation model under Few-shot CoT. We present the results in Figure 3. Across the three tasks, ReFeri yields consistent improvements as K increases. On *MATH500*, while the accuracy of random selection decreases as the number of generated samples increases, ReFeri consistently selects higher-quality responses, improving from 75.8% at $K = 1$ to 79.2% at $K = 15$. On *GPQA*, where ReFeri raises performance from 41.4% to 45.5% as the candidate pool grows. Consistently, the largest gain is observed on *MuSR-ta*, which saw a sharp jump in accuracy from 75.6% to 86.0%, an improvement of 10.4%. In contrast, while USC is effective on *MATH500*, its performance even degrades with more candidates on *GPQA* and *MuSR-ta*, where candidate responses are more diverse and LLM has relatively less relevant knowledge. In addition, we observe an inherent ordering bias in USC: over 90% of

selections come from the first two responses (see Appendix B.3), indicating a limitation of this prompt-base approach. Overall, these results confirm that ReFeri scales well with more candidates, demonstrating effectiveness and reliability in practical test-time scaling.

4.3. Additional analyses

In this section, we conduct additional analyses of ReFeri. We mainly conduct experiments using GPT-4o-mini on *MATH500*, *GPQA*, and *MuSR-ta*. More results are presented in Appendix B.

Ablation study. To better understand which components contribute to the effectiveness of ReFeri, we conduct an ablation study on each part of the proposed scoring function (Eq. 10). We focus on two primary components: the forward confidence score (Eq. 6) and the backward consistency score (Eq. 8), both of which are grounded in Bayes’ rule (Eq. 5). The results are presented in Table 2. First, it is observed that combining forward and backward scores outperforms either score alone, validating the joint consideration. This complementary effect is from their different nature; while forward score focuses on model-generated response which may contain noise, backward score utilizes given few-shot examples which are well-curated inputs and ground-truth labels, and thus relatively clean. Accordingly, the forward and backward scores anchored on few-shot examples form the structural backbone of our method. Interestingly, adding the direct estimation like Eq. 10 consistently improves accuracy across benchmarks, though it is not required by the Bayes formulation. This result suggests its utility as a supplementary prior for verifying LLM outputs. More results are in Appendix B.

Estimation models and few-shot examples. To examine whether ReFeri is sensitive to the choice of estimation model p_θ , we evaluate its performance using three LLMs with diverse scales and architectures: LLaMA-3.2-1B-Instruct (MetaAI, b), Qwen-2.5-7B-Instruct (Qwen), and LLaMA-3.1-70B-Instruct (MetaAI, a). The generation model is fixed (either GPT-4o-mini, GPT-4o, or LLaMA-3.1-8B), and we apply each estimation models to two tasks on *MATH500* and *GPQA*. The average accuracy of three generation LLMs is presented in Figure 4 (Full results are in Appendix B). Here, ReFeri consistently improves Few-shot CoT across all settings, with an average gain of 5.0% on *MATH500* and 5.3% on *GPQA*. Notably, the smallest model (LLaMA-3.2-1B) performs competitively, and even achieves competitive performance on *MATH500*. We attribute this to the relative simplicity of *MATH* benchmark, as recent small LLMs often exhibit reasonable performance; hence, they can make reliable likelihood estimates for selection. In contrast, *GPQA*

Table 1. Main Results. Overall performance on seven reasoning benchmarks comparing the proposed **ReFeri** with different baselines not require additional training, under three different state-of-the-art LLMs. The best and second-best scores are highlighted in **bold** and underline, respectively.

Models	Methods	MATH500 (Acc.)	MMLU-pro (Acc.)	GPQA (Acc.)	DROP (EM / F1)	HotpotQA (EM / F1)	MuSR-op (Acc.)	MuSR-ta (Acc.)	Avg.
GPT-4o-mini	Zero-shot CoT	76.4	63.0	<u>43.0</u>	77.6 / <u>85.6</u>	31.5 / 41.4	58.1	56.2	58.0
	Few-shot CoT	75.2	63.0	41.3	76.8 / 83.1	34.0 / 45.1	59.4	77.0	61.0
	USC	78.6	62.5	42.4	<u>78.8</u> / 85.8	36.6 / 48.2	<u>59.8</u>	76.4	<u>62.2</u>
	CoT-WP	77.8	<u>64.2</u>	42.4	77.6 / 83.1	34.0 / <u>45.6</u>	57.0	<u>79.6</u>	61.8
	LEAP	74.5	63.2	43.9	75.8 / 83.0	34.0 / 45.1	<u>59.8</u>	74.4	60.8
	ReFeri (Ours)	<u>78.2</u>	65.0	42.4	79.6 / 85.3	<u>36.2</u> / <u>47.9</u>	61.3	82.8	63.6
GPT-4o	Zero-shot CoT	77.5	73.9	48.8	75.1 / 85.3	37.6 / 49.9	61.7	66.6	63.0
	Few-shot CoT	75.6	73.7	47.8	80.6 / 89.2	44.6 / 58.4	69.7	87.0	68.4
	USC	79.8	72.1	<u>50.5</u>	82.0 / 90.2	45.8 / 60.4	73.4	87.6	<u>70.2</u>
	CoT-WP	78.0	<u>75.0</u>	48.5	<u>83.0</u> / <u>90.8</u>	48.0 / 61.2	70.3	<u>88.8</u>	<u>70.2</u>
	LEAP	75.6	74.0	45.5	81.5 / 89.8	45.1 / 58.4	66.8	87.2	68.0
	ReFeri (Ours)	<u>78.4</u>	75.5	51.5	83.6 / 91.2	<u>47.4</u> / <u>61.0</u>	<u>72.3</u>	90.8	71.4
LLaMA-3.1-8B	Zero-shot CoT	44.2	39.8	21.6	60.4 / 66.4	15.2 / 21.2	50.6	43.0	39.3
	Few-shot CoT	42.9	38.7	24.0	61.4 / 67.3	19.0 / 25.1	53.3	64.8	43.1
	USC	49.6	35.6	28.8	69.6 / <u>75.8</u>	24.4 / <u>32.5</u>	52.3	67.2	46.8
	CoT-WP	47.8	<u>44.8</u>	<u>32.3</u>	70.2 / 75.1	25.0 / 32.2	56.6	<u>71.6</u>	<u>49.8</u>
	LEAP	42.3	37.3	27.8	58.2 / 64.1	19.9 / 26.8	51.6	69.2	43.8
	ReFeri (Ours)	51.0	45.1	34.8	70.2 / 76.7	25.0 / 33.0	56.6	80.0	51.8

Table 2. Ablation study. Evaluation of scoring variants on responses by GPT-4o-mini, comparing the contribution of metric term (forward, backward, and direct) on MATH500 and GPQA.

	Forw.	Back.	Dire.	MATH500	GPQA
	✗	✗	✗	75.2	41.3
	✓	✗	✗	77.6	41.9
	✗	✓	✗	75.2	41.9
	✓	✓	✗	77.8	42.4
ReFeri	✓	✓	✓	78.2	42.4

Table 3. Different few-shot examples. Accuracy across three different choices of few-shot examples on MATH500 (top) and GPQA (bottom) using GPT-4o-mini to generate responses.

Methods	1st	2nd	3rd	Avg.
Few-shot CoT	74.4	73.8	74.7	74.3
ReFeri (Ours)	76.8	77.4	79.0	77.7
Few-shot CoT	41.7	39.7	44.8	42.1
ReFeri (Ours)	45.5	41.4	46.0	44.3

requires more complex reasoning; therefore, using the large estimation model could be more beneficial. Indeed, LLaMA-3.1-70B achieves the best performance on this case. 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 estimation model.

Also, ReFeri highly relies on few-shot examples for scoring of both forward and backward scores (Section 3.2). This raises the question of how sensitive the method is to the choice of few-shot exemplars. To this end, we conduct a sensitivity study on MATH500 and GPQA using GPT-4o-mini, where we randomly sample three different few-shot examples from the training dataset. As shown in Table 3, both Few-shot CoT and ReFeri show some variation across seeds. Nevertheless, ReFeri consistently outperforms Few-shot CoT which corresponded to random selection, and the average gap remains approximately 2.8%. These results indicate that ReFeri remains robust to exemplar choice and is consistently effective, rather than overfitted to specific demonstrations.

Generalization to reasoning LLM. As discussed in Section 2, few-shot prompting is often less effective for RL-tuned LLMs such as DeepSeek-R1 (Guo et al., 2025). This degradation may stem from several factors, including sensitivity to prompt formatting, incompatibility with few-shot templates, or reliance on special tokens (e.g., `<think>`). We

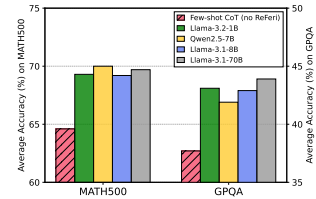


Figure 4. Estimation model. Each bar shows the average accuracy of three generation LLMs on MATH500 and GPQA.

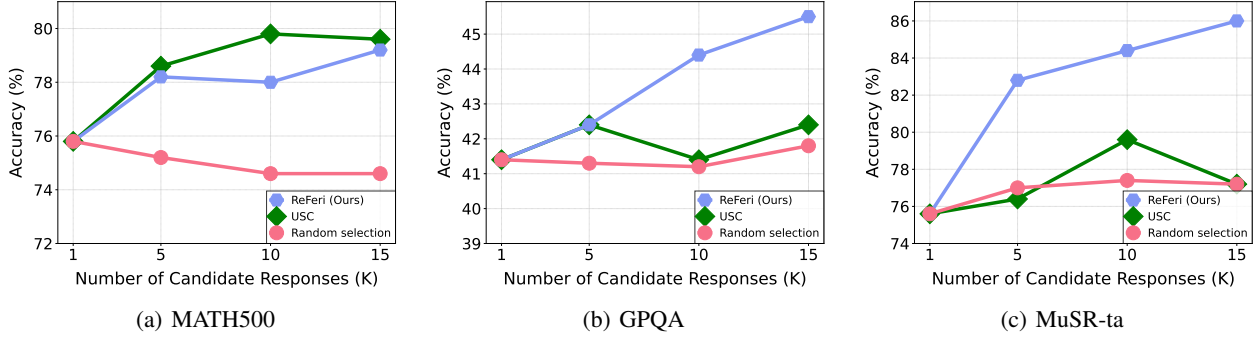


Figure 3. **Test-time scaling with ReFeri.** Accuracy of ReFeri versus other training-free selection methods (Random selection and USC) on MATH500, GPQA, and MuSR-ta. GPT-4o-mini generate different numbers of candidate responses ($K = 1, 5, 10, 15$) using Few-shot CoT, similar to Table 1. Solid lines indicate the performance of ReFeri (ours), USC, and Random, respectively.

Table 4. **Generalization to reasoning LLM.** Accuracy of DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) under zero-shot CoT and few-shot CoT methods with or without ReFeri.

Methods	AIME24	AMC23
Zero-shot CoT	52.7	84.5
+ ReFeri (Ours)	63.3	90.0
Few-shot CoT	44.0	78.0
+ ReFeri (Ours)	50.0	85.0

also observe similar results through experiments on two challenging benchmarks, *AIME24* (aim) and *AMC23* (amc), with DeepSeek-R1-Distill-Qwen-7B (DeepSeek); for the experiments, we generate five reasoning paths under a temperature of 1.0 with five few-shot examples from MATH500, following (Yang et al., 2024). We use this model for both generation and estimation. More detailed setups are presented in Appendix A.3. As shown in Table 4, Zero-shot CoT achieves 52.7% and 84.5% respectively, while Few-shot CoT yields a notable drop to 44.0% (-8.7%) and 78.0% (-6.5%). Nevertheless, the proposed ReFeri successfully selects the promising reasoning path and yields the significant accuracy gain (+6.0% and +7.0%, respectively). This result confirms the robustness of ReFeri from the utilization of well formatted few-shot examples for the validation.

On the other hand, one can observe that the final accuracy is still lower than the average accuracy by Zero-shot CoT, mainly due to the limited accuracy of reasoning paths from Few-shot CoT. However, as described in Eq. 3, ReFeri is indeed applicable to select reasoning paths of zero-shot CoT, while we primarily apply to few-shot CoT as it usually yields better reasoning paths (Table 1). With the experiments, we verify that applying ReFeri to Zero-shot CoT yields substantial improvements that raise accuracy to 63.3% (+10.6%) on AIME24 and 90.0% (+5.5%) on AMC23, respectively. These results are evidence that few-shot examples in ReFeri mainly serve as a form of post-hoc validation pipeline, not the generation guidance like conventional Few-shot CoT.

Table 5. **LLM personalization.** Evaluation results on LaMP-4 and LaMP-5 using GPT-4o-mini as generator. *Vanilla* uses no history, while *Few-shot RAG* retrieves user history via BM25.

Methods	LaMP-4		LaMP-5	
	Rouge-1	Rouge-L	Rouge-1	Rouge-L
Vanilla	0.120	0.106	0.421	0.332
Few-shot RAG	0.142	0.126	0.453	0.368
ReFeri (Ours)	0.160	0.141	0.503	0.402

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.

Application to LLM personalization. Lastly, we further apply ReFeri for *LLM personalization* to evaluate its broader applicability. The goal of LLM personalization is steering LLMs’ responses towards the individual users, which becomes progressively important (Salemi et al., 2024; Tan et al., 2024; Kim & Yang, 2025). One representative baseline for LLM personalization is few-shot retrieval-augmented generation (RAG) that retrieved the user’s previous data relevant to the given test query, and hence it’s natural to apply ReFeri. Specifically, we evaluate on two tasks in *LaMP* benchmark (Salemi et al., 2024), LaMP-4 (personalized news headline generation) and LaMP-5 (personalized scholarly title generation), and use GPT-4o-mini as generation LLM. We generate $K = 5$ candidate responses with a temperature of 1.0 as same as Table 1. *Vanilla* baseline directly answers to query without external context, while the *Few-shot RAG* baseline augments input prompt with $N = 5$ examples retrieved via BM25 (Robertson et al., 2009) from the user’s history. Following (Salemi et al., 2024), we evaluate all responses against gold references using ROUGE-1 and ROUGE-L. The average of all K responses is reported for the baselines, and results with the selected response is reported for ReFeri, respectively. As shown in Table 5, ReFeri consistently outperforms both baselines across LaMP-4 and

LaMP-5. Notably, it improves ROUGE-L from 0.368 to 0.402 on LaMP-5, and from 0.126 to 0.141 on LaMP-4. This result demonstrates the applicability of ReFeri beyond traditional reasoning tasks—to open-ended, user-specific scenarios.

5. Conclusion

We propose ReFeri, a training-free framework for selecting promising output from LLM that reuses few-shot examples not only for generation but also for validation. In our experiments, ReFeri performs consistently effective in various LLMs and tasks, demonstrating strong adaptability to RL-tuned LLMs and open-ended personalization settings, and highlighting its generality beyond reasoning tasks. Our results suggest that ReFeri is a practical way to find the reliable LLM output with minimal human involvement, opening future directions to reconsider the broader utility of demonstrations.

Limitation and future works. Since the selection by ReFeri is determined by likelihoods produced by an estimation model, it does not explain why a response is incorrect, unlike PRMs, which offer step-level feedback, or LLM-as-judge, which can easily generate explanations by prompting. Additionally, performing multiple inferences with the estimation model to obtain likelihoods for different few-shot examples (Eq. 8) can be costly, although it is still more efficient than generating responses with a state-of-the-art LLM like GPT-4o. However, as shown in Table 4, smaller estimation models can be highly effective depending on the task and domain; consequently, adaptively using models of varying scales of the estimation model could be effective in solving this issue.

Impact Statement

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 personalization tasks with limited access to user 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.

References

Aime24. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024.

Amc23. <https://huggingface.co/datasets/zwe99/amc23>.

Aggarwal, P., Madaan, A., Yang, Y., et al. 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)*, 2023.

Anthropic. Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chen, X., Aksitov, R., Alon, U., Ren, J., Xiao, K., Yin, P., Prakash, S., Sutton, C., Wang, X., and Zhou, D. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek. Deepseek-r1-distill-qwen-7b. <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B>.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. 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)*, 2019.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Fu, Y., Peng, H., Sabharwal, A., Clark, P., and Khot, T. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*, 2023.

Ge, Y., Liu, S., Wang, Y., Mei, L., Chen, L., Bi, B., and Cheng, X. Innate reasoning is not enough: In-context learning enhances reasoning large language models with less overthinking. *arXiv preprint arXiv:2503.19602*, 2025.

Gui, L., Gârbaacea, C., and Veitch, V. Bonbon alignment for large language models and the sweetness of best-of-n sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

- Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.
- Huang, X., Zhang, L. L., Cheng, K.-T., Yang, F., and Yang, M. Fewer is more: Boosting math reasoning with reinforced context pruning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.
- Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.
- Jin, M., Yu, Q., Shu, D., Zhao, H., Hua, W., Meng, Y., Zhang, Y., and Du, M. The impact of reasoning step length on large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- Kim, J. and Yang, Y. Few-shot personalization of llms with mis-aligned responses. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2025.
- Kim, J., Nam, J., Mo, S., Park, J., Lee, S.-W., Seo, M., Ha, J.-W., and Shin, J. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. In *International Conference on Learning Representations (ICLR)*, 2024.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let’s verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.
- Mahan, D., Van Phung, D., Rafailov, R., Blagden, C., Lile, N., Castriato, L., Fränken, J.-P., Finn, C., and Albalak, A. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.
- MetaAI. Llama-3.1-70b-instruct. <https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct>, a.
- MetaAI. Llama-3.2-1b-instruct. <https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct>, b.
- Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.
- OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024a.
- OpenAI. Gpt-4o mini: advancing cost-efficient intelligence. <https://platform.openai.com/docs/models/gpt-4o-mini/>, 2024b.
- OpenAI. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>, 2024c.
- Peng, K., Ding, L., Yuan, Y., Liu, X., Zhang, M., Ouyang, Y., and Tao, D. Revisiting demonstration selection strategies in in-context learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- Perez, E., Kiela, D., and Cho, K. True few-shot learning with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Qiu, X. and Mikkilainen, R. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- Qu, Y., Zhang, T., Garg, N., and Kumar, A. Recursive introspection: Teaching language model agents how to

- self-improve. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- Qwen. Qwen2.5-7b-instruct. <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>.
- Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- Robertson, S., Zaragoza, H., et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- Salemi, A., Mysore, S., Bendersky, M., and Zamani, H. Lamp: When large language models meet personalization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Sprague, Z., Ye, X., Bostrom, K., Chaudhuri, S., and Durrett, G. Musr: Testing the limits of chain-of-thought with multistep soft reasoning, 2024. In *International Conference on Learning Representations (ICLR)*, 2024.
- Sprague, Z., Yin, F., Rodriguez, J. D., Jiang, D., Wadhwa, M., Singhal, P., Zhao, X., Ye, X., Mahowald, K., and Durrett, G. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Tan, Z., Zeng, Q., Tian, Y., Liu, Z., Yin, B., and Jiang, M. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.
- Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.
- Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a.
- Wang, X. and Zhou, D. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Wu, Z., Wang, Y., Ye, J., and Kong, L. 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)*, 2023.
- Yan, K., Xu, Y., Du, Z., Yao, X., Wang, Z., Guo, X., and Chen, J. Recitation over reasoning: How cutting-edge language models can fail on elementary school-level reasoning problems? *arXiv preprint arXiv:2504.00509*, 2025.
- Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023. In *International Conference on Learning Representations (ICLR)*, 2023.
- Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. E. Self-rewarding language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

- Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 2025.
- Zhang, T., Madaan, A., Gao, L., Zheng, S., Mishra, S., Yang, Y., Tandon, N., and Alon, U. In-context principle learning from mistakes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

A. More Details of Experimental Setups

This section covers more details about the experiments from Section 4.

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 (Yang et al., 2024)⁴ for GPT-based models and (Lewkowycz et al., 2022)⁵ 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., 2024b) 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 follow the format used in (Wang et al., 2024b). This benchmark is used 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 release (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., 2022) 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). 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 256 TeamAllocation and 250 ObjectPlacement examples from the official release (Sprague et al., 2024), using 3-shot prompts tailored to each task (Sprague et al., 2025). MuSR requires understanding of narrative flow, contextual logic, and physical feasibility. As demonstrated in (Sprague et al., 2025), ICL plays a critical role in model performance on MuSR, and demonstrates the effectiveness of ReFeri.
- **AIME24.** AIME24 (aim) consists of 30 official problems from the 2024 AIME I and II exams, widely known for their difficulty in symbolic math reasoning. These integer-answer questions cover topics such as algebra, combinatorics,

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

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

and number theory, and are commonly used to evaluate mathematical reasoning capabilities of RLLMs. We use all 30 problems as-is and apply the same 5-shot prompting setup as in MATH500, following the implementation of (Yang et al., 2024).

- **AMC23.** AMC23 (amc) consists of 40 problems and includes selected high-difficulty problems from the 2023 AMC 12A and 12B exams. Evaluation is performed using the same 5-shot prompting setup as MATH500, following the implementation of (Yang et al., 2024). This benchmark provides additional resolution for evaluating symbolic reasoning below the AIME level.
- **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.

A.2. Baselines

Here, we provide the template used for our baseline, using MATH500 as a representative task among multiple benchmarks. (see list 1–7).

A.3. Implementation

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

Resource details. To avoid out-of-memory, we used two NVIDIA H100 GPUs for evaluation with the LLaMA-3.1-70B-Instruct model, and one H100 for DeepSeek-R1-Distill-Qwen-7B. All other experiments were performed on a single A6000 GPU.

Response generation. We use lm-eval-harness⁶ to generate responses from LLaMA-based models, with temperature set to 1.0 and 5 responses sampled per input. The prompt was written in chat template format and used vllm.⁷ 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, we report the average score across the 5 generations. 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. Except Table 4, we generate multiple candidate responses $\{r_1, \dots, r_K\}$ for each test query using Few-shot CoT, as it exhibit the better quality on the average (see Table 1).

Generalization to reasoning LLM. We evaluate DeepSeek-R1-Distill-Qwen-7B (DeepSeek) on two challenging math benchmarks: AMC23 and AIME24. For both datasets, we adopt the same 5-shot prompting used in MATH500, following the implementation of (Yang et al., 2024), as no task-specific few-shot examples are readily available and most prior works tend to prefer zero-shot formats due to frequent performance degradation under few-shot prompting (Guo et al., 2025). We perform 5 generations per example using nucleus sampling ($p = 0.95$), and temperature 1, running vLLM with two H100 GPUs, with a maximum input and output length of 32,768 tokens. Our implementation is adapted from the FuseAI framework.⁸ We report the average of the five responses in the case of Zero-shot CoT and Few-shot CoT. To apply ReFeri, we follow the same procedure as in all other experiments. When a zero-shot setting is required, we simply substitute the model’s zero-shot response r_k for each few-shot response in the same pipeline.

Application to LLM personalization. We experiment with two configurations: (1) *vanilla generation*, where only the user query is used as input without any personalization; and (2) *RAG-based personalization*, where the prompt is augmented with both a user profile and top-5 similar examples retrieved from the user’s history using BM25 (Salemi et al., 2024; Tan et al., 2024). The responses are generated using the GPT-4o-mini model via the OpenAI API with temperature set to 1.0.

Building on the outputs generated through above pipeline, we apply our ReFeri method to select the most likely response

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

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

⁸https://github.com/fanqiwang/FuseAI/tree/main/Fuse01-Preview/math_evaluation

Algorithm 1 ReFeri algorithm

Input: estimation model P , test-query \tilde{q} , K candidate responses $\{r_1, \dots, r_K\}$, N few-shot examples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, replaced prompt $\tilde{\mathbf{X}}_i$, hyperparameter β

```

1: for  $k = 1$  to  $K$  do
2:    $S_{\text{Forw}} \leftarrow$  Compute forward score with  $r_k$  as label, using  $P$  and context  $(\tilde{q}, \mathbf{X})$  (Eq. (6))
3:    $S_{\text{Dire}} \leftarrow$  Compute direct score with  $r_k$  as label, using  $P$  and only  $\tilde{q}$ 
4:   Initialize backward score  $S_{\text{Back}} = 0$ 
5:   for  $i = 1$  to  $N$  do
6:     Construct  $\tilde{\mathbf{X}}_i \leftarrow$  using a leave-one-out strategy (Eq. (7))
7:      $S_{\text{Back}} \leftarrow S_{\text{Back}} +$  backward score with  $a_i$  as label, using  $P$  and  $\tilde{\mathbf{X}}_i$  (Eq. (8))
8:   end for
9:    $S_{\text{Comb}} \leftarrow S_{\text{Forw}} - S_{\text{Back}}$  (Eq. (9))
10:   $S_{\text{Fin}} \leftarrow (1 - \beta) \cdot S_{\text{Dire}} + \beta \cdot S_{\text{Comb}}$  (Eq. (10))
11:   $S_k \leftarrow S_{\text{Fin}}$ 
12: end for
13:  $r_{k^*} \leftarrow \arg \max_k S_k$  (Eq. (3))
14: return  $r_{k^*}$ 

```

among the five candidates for each input, with $\beta \in \{0.5, 1\}$. For the evaluation, the ROUGE score between the response and the gold reference is used following prior works. Here, we observe that $\beta = 1$ consistently yields better performance than $\beta = 0.5$, which indicates the ineffectiveness of direct estimation in this task. This selection strategy shows improved alignment with the gold answers and performs favorably compared to vanilla generation and RAG-based personalization (see Table 5).

B. More Quantitative Results

B.1. Full results with different estimation models

Table 6. Full results with different estimation models on MATH500 and GPQA

(a) MATH500					(b) GPQA				
Estimation	GPT-4o-mini	GPT-4o	LLaMA3.1-8B	Avg	Estimation	GPT-4o-mini	GPT-4o	LLaMA3.1-8B	Avg
LLaMA-3.2-1B	78.0	77.8	52.0	69.3	LLaMA-3.2-1B	44.9	50.0	34.3	43.1
Qwen-2.5-7B	78.8	79.0	52.2	70.0	Qwen-2.5-7B	41.4	50.5	33.8	41.9
LLaMA-3.1-8B	78.2	78.4	51.0	69.2	LLaMA-3.1-8B	42.4	51.5	34.8	42.9
LLaMA-3.1-70B	77.8	78.0	53.4	69.7	LLaMA-3.1-70B	42.9	53.5	35.4	43.9

Table 6 provides full results for all estimation model combinations of MATH500 and GPQA. This complements the average performance across different generation LLMs (GPT-4o-mini, GPT-4o, and LLaMA3.1-8B) shown in Figure 4. On both tasks, ReFeri shows consistent performance gains regardless of the estimation model used, emphasizing its robustness. There are some model-specific trends; for example, smaller models (LLaMA-3.2-1B) perform competitively on (relatively) simple tasks like MATH500, as discussed in Section 4.2.

B.2. Additional ablation

Here, we conduct the additional experiments to provide comprehensive ablation study for ReFeri. First, 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 score, 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_k as additional context. Specifically, this variant modifies the backward score in Eq. (8) by replacing the leave-one-out prompt $\tilde{\mathbf{X}}_i$ with a single pair $\mathbf{y}_k = (\tilde{q}, r_k)$:

$$S'_{\text{Back}}(r_k) := \log P(\mathbf{X} \mid \mathbf{y}_k) - \log P(\mathbf{X}) = \sum_{i=1}^N [\log P(a_i \mid q_i, \tilde{q}, r_k) - \log P(a_i \mid q_i)],$$

This formulation can be interpreted as the most straight-forward implementation of backward score (see Eq. 5) under the assumption of mutual independence between few-shot examples. As shown in Table 7, the accuracy under *No replace* is consistently less or equal than ReFeri (6 of 7). We attribute this to the fact that using full leave-one-out prompts better reflects the consistency of \mathbf{y}_k with the original in-context reasoning trajectory. Nonetheless, *No replace* could serve as a practical alternative that trades off a small performance drop with the greater simplicity.

Next, we also examine the effectiveness of the direct likelihood term $\log P(\mathbf{y}_k) := \log P(r_k | \tilde{q})$ when used alone (*Direct*). While this term is not required by the Bayes formulation, we initially hypothesized that it may serve as a useful prior when combined with other terms. However, as shown in Table 7, relying solely on the direct term yields lower accuracy (63.2%) than our scoring approach (63.6%). Remarkably, *Direct* slightly outperforms on MATH500 and GPQA, which are frequently encountered and handled domains (math and science), but it significantly underperforms on other domains which LLM should be adapted through in-context learning. This result supports our motivation of additional utilization of few-shot examples for the selection, to complement LLM’s limited intrinsic knowledge, especially for a new domain, through few-shot examples.

Table 7. **Additional ablation study.** Evaluation of scoring variants on responses by GPT-4o-mini on MATH500 and GPQA.

Methods	MATH500 (Acc.)	MMLU-pro (Acc.)	GPQA (Acc.)	DROP (EM / F1)	HotpotQA (EM / F1)	MuSR-op (Acc.)	MuSR-ta (Acc.)	Avg.
No replace	78.4	65.0	42.4	78.4 / 84.1	36.2 / 48.0	60.2	82.4	63.3
Direct	78.4	64.9	42.9	79.2 / 84.9	35.2 / 47.1	59.8	82.0	63.2
ReFeri (Ours)	78.2	65.0	42.4	79.6 / 85.3	36.2 / 47.9	61.3	82.8	63.6

B.3. Additional comparison with few-shot prompting-based selection methods

Table 8. **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.

Models	Methods	MATH500 (Acc.)	MMLU-pro (Acc.)	GPQA (Acc.)	DROP (EM / F1)	HotpotQA (EM / F1)	MuSR-op (Acc.)	MuSR-ta (Acc.)	Avg.
GPT-4o-mini	USC	78.6	62.5	42.4	78.8 / 85.8	36.6 / 48.2	59.8	76.4	62.2
	USC-w/ Fewshot	79.4	62.9	38.9	77.8 / 85.5	36.2 / 48.3	59.0	77.2	61.6
	LLM-as-Judge	79.0	64.7	39.9	78.6 / 85.6	36.2 / 48.4	58.6	77.2	62.0
	ReFeri (Ours)	78.2	65.0	42.4	79.6 / 85.3	36.2 / 47.9	61.3	82.8	63.6
GPT-4o	USC	79.8	72.1	50.5	82.0 / 90.2	45.8 / 60.4	73.4	87.6	70.2
	USC-w/ Fewshot	80.2	72.2	47.5	82.6 / 90.7	46.2 / 60.7	71.1	89.6	69.9
	LLM-as-Judge	80.8	76.6	51.0	82.4 / 90.2	46.6 / 61.3	72.3	89.2	71.3
	ReFeri (Ours)	78.4	75.5	51.5	83.6 / 91.2	47.4 / 61.0	72.3	90.8	71.4
LLaMA-3.1-8B	USC	49.6	35.6	28.8	69.6 / 75.8	24.4 / 32.5	52.3	67.2	46.8
	USC-w/ Fewshot	47.8	36.5	28.3	69.0 / 75.3	25.2 / 32.3	53.9	70.0	47.2
	LLM-as-Judge	46.0	44.1	21.2	67.6 / 74.0	23.4 / 31.2	55.1	66.4	46.2
	ReFeri (Ours)	51.0	45.1	34.8	70.2 / 76.7	25.0 / 33.0	56.6	80.0	51.8

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).

As shown in Table 8, 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

Table 9. Response selection distribution per task (GPT-4o-mini).

Task	Method	#1	#2	#3	#4	#5	Fail (-1)
MATH500	USC	87.8	9.4	1.0	0.4	1.4	0.0
	USC-w/ Fewshot	89.2	8.6	0.8	0.2	1.2	0.0
	LLM-as-Judge	91.0	4.6	1.0	2.4	1.0	0.0
MMLU-Pro	USC	73.2	22.2	1.1	1.4	2.1	0.3
	USC-w/ Fewshot	73.7	20.9	1.6	1.2	2.5	0.3
	LLM-as-Judge	68.2	12.6	8.3	4.7	4.7	1.6
GPQA	USC	66.2	29.3	0.5	0.0	4.0	0.0
	USC-w/ Fewshot	65.7	25.3	3.0	1.0	5.1	0.0
	LLM-as-Judge	37.4	26.3	16.7	10.6	5.1	4.0
DROP	USC	81.8	15.4	1.2	0.6	1.0	0.0
	USC-w/ Fewshot	82.8	13.8	1.2	0.8	1.4	0.0
	LLM-as-Judge	85.0	9.2	2.2	2.8	0.8	0.0
HotpotQA	USC	52.6	33.6	7.4	3.8	2.6	0.0
	USC-w/ Fewshot	65.4	23.0	5.4	3.2	3.0	0.0
	LLM-as-Judge	62.2	16.6	8.6	8.0	4.6	0.0
MuSR-op	USC	91.4	7.42	0.4	0.4	0.4	0.0
	USC-w/ Fewshot	92.6	7.42	0.0	0.0	0.0	0.0
	LLM-as-Judge	85.2	7.81	3.9	1.2	2.0	0.0
MuSR-ta	USC	55.6	31.2	2.4	1.2	0.0	9.6
	USC-w/ Fewshot	60.8	33.2	6.0	0.0	0.0	0.0
	LLM-as-Judge	34.4	36.4	272	0.0	2.0	0.0

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, over 90% of USC selections were made from the first two responses regardless of correctness (see Table 9). This highlights a critical weakness in prompt-based selection: the output is often determined more by position than content. In contrast, our approach mitigates such ordering artifacts by decoupling few-shot demonstrations 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) or smaller models. 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.

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.

C.1. Response level analysis

For each response selected by ReFeri, we compute the Eq. (10) with $\beta = 1$ 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. As shown in below examples, the selected response by ReFeri has more accurate reasoning. For instance, in the example of MATH-500, the best-scoring response anchor 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 combine 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 are consistent with HotpotQA’s multi-hop requirements by making intermediate hops explicit and factually correct, while rejected answer provide unsupported single-hop claims.

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]
```

Responses for MATH500:

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 (width) \times 4 (height) – 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 (3, 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.

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 ^1H 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 ^1H NMR spectrum, none of which correspond to vinyl-hydrogens. One of the signals in the ^1H 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. **¹H 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 ¹H 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) CH3C(H)(CH3)C(H)(CH3)CH2COOH: A branched structure likely leading to complex splitting.
- (B) CH3CH2C(H)(CH3)C(H)(CH3)COOH: A linear structure with groups that may contribute to the complexity in splitting.
- (C) CH3C(H)(C2H5)C(H)(C2H5)CH2COOH: Also branched, with ethyl groups that could lead to intricate splitting.
- (D) CH3CH2C(H)(C2H5)C(H)(C2H5)COOH: A linear structure including ethyl groups, likely resulting in simpler splitting.

4. **Choosing the Compound**:

- The broad absorption at 3000 cm⁻¹ and significant peaks at 1700 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).

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

Responses for HotpotQA:

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  $\frac{\frac{1}{3}(1-(\frac{2}{3})^5)}{1-\frac{2}{3}} = \boxed{\frac{211}{243}}$ .
The answer is  $\frac{211}{243}$ 

...

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}
'''
```