Mitigating Demonstration Bias through Global Coevolutionary Reasoning

Anonymous ACL submission

Abstract

Recent advances in large language models (LLMs) have demonstrated the effectiveness of chain-of-thought (CoT) prompting. Few-Shot-CoT relies on task-specific, manually labeled demonstrations, limiting its generalization to unseen tasks. While Zero-Shot-CoT eliminates this reliance, it often underperforms. To address this, existing methods aim to automatically generate demonstrations in zero-shot settings. However, these generated demonstrations face challenges due to demonstration bias: 1) selected demonstrations may contain errors, and 2) they may not be suitable or representative enough for all questions. To mitigate these biases, we propose Global Coevolutionary Reasoning (GCR). The method first applies Zero-Shot-CoT to answer all questions, then clusters the results. For each cluster, a random sample is selected, and these selected samples serve as demonstrations for each other. The model then iteratively re-answers the questions and updates their rationales based on these demonstrations, enabling coevolutionary reasoning to progressively improve the quality of the answers. This process of random sampling and coevolutionary reasoning is repeated until all questions have been re-answered. Experimental results on ten datasets using GPT-3.5-turbo and GPT-4o-mini show that GCR outperforms baseline methods without any performance degradation caused by demonstration bias. Additionally, GCR is orthogonal to existing methods and can be seamlessly integrated with them.

1 Introduction

The paradigm of in-context learning (ICL) (Brown et al., 2020) has proven effective in large language models (LLMs), enabling them to perform reasoning tasks based on a few examples. Among the strategies within ICL, chain-of-thought (CoT) (Wei et al., 2022) reasoning, including few-shot and zero-shot variants, is a cornerstone for tackling complex problems. Few-Shot-CoT relies on manually labeled demonstrations, limiting generalization to unseen tasks, while Zero-Shot-CoT (Kojima et al., 2022) removes this reliance, offering broader applicability but often resulting in suboptimal reasoning.

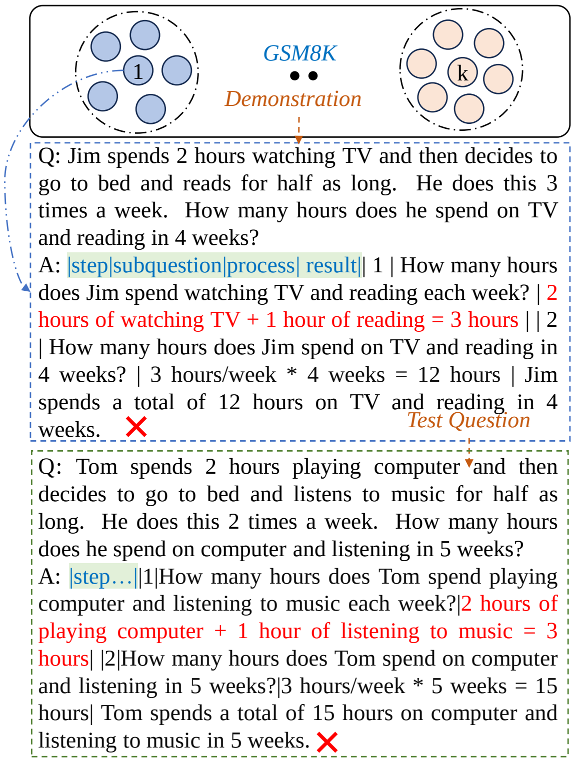

Figure 1: An example of the first type of demonstration bias in GPT-3.5-turbo with Auto-CoT on the GSM8K dataset, where the selected demonstrations may contain errors and propagate incorrect reasoning.

The Auto-CoT (Zhang et al., 2023) framework addresses these challenges by dynamically transitioning from zero-shot to Few-Shot-CoT. It clusters questions based on semantic similarity and selects representative examples from the cluster centroids to construct a few-shot prompt, combining the strengths of both approaches. However, Auto-CoT faces demonstration bias: 1) selected

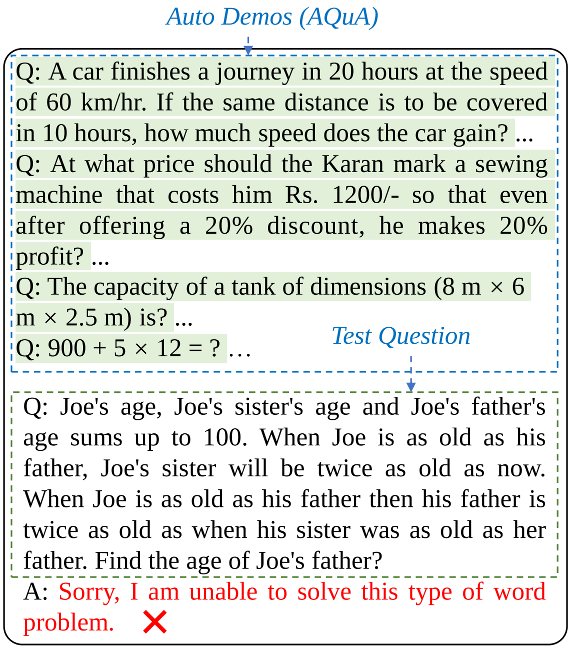

Figure 2: An example of the second type of demonstration bias in GPT-3.5-turbo with Auto-CoT on the AQuA dataset, where the selected demonstrations may be unrelated to the question being answered, leading to incorrect answers.

demonstrations may contain errors, propagating incorrect reasoning (Figure 1), and 2) they may not be representative of all questions (Figure 2). While increasing demonstration diversity can mitigate the effect of erroneous demonstrations, it does not fully resolve the first bias.

ECHO (Jin and Lu, 2024) attempts to iteratively refine demonstrations by using each as a demonstration for the others, fostering coevolutionary reasoning between demonstrations to eliminate the first bias. However, this iterative process lacks a mechanism to ensure positive progression, potentially exacerbating the first bias (Figure 3), and does not address the second bias related to representativeness.

Coevolutionary reasoning posits that **good demonstrations are more likely to guide LLMs to correct answers**. Demonstrations are considered "good" when they are both representative and accurate. In Shtok et al. (2024), representativeness is defined as the proportion of questions in a set that can be solved using a specific example as a demonstration, with the set's problem-solving success rate exceeding a threshold. However, obtaining such problem sets with known correct answers in practice is difficult, limiting the applicability of this approach. Auto-CoT clusters questions and selects

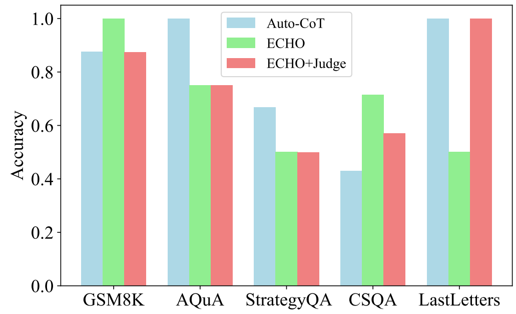

Figure 3: Comparison of demonstration accuracy across Auto-CoT, ECHO, and ECHO+Judge. ECHO can improve demonstration accuracy in some cases but may also degrade it in others. On the other hand, ECHO+Judge, which incorporates a judging mechanism during iterative updates and is influenced by judge bias, can enhance accuracy in certain situations while occasionally reducing it.

centroids as demonstrations, assuming these are representative. However, semantic similarity does not necessarily imply similar reasoning methods (Xu et al., 2024b; An et al., 2023), and centroids may not adequately represent points that are far from them. (Figure 2). Zhang and Ding (2024) emphasizes the need to explore the prompt space in addition to the answer space. While traditional methods (Yao et al., 2024; Besta et al., 2024) adopt a "one prompt for all" approach, a single prompt or set of demonstrations cannot effectively address all questions. As noted in Yuan et al. (2024), finding a universally applicable prompt for every question remains a challenging task. The ideal solution would involve designing a specific prompt for each question, but manually creating such prompts is impractical. Thus, **coevolutionary reasoning focuses on optimizing demonstrations for greater accuracy**. While manually created demonstrations do not require validation of their correctness, constructing task-specific demonstrations for new tasks is often time-consuming and labor-intensive. This has led to the consideration of automatically generated demonstrations. However, without external feedback, these auto-generated demonstrations are prone to errors. For instance, Auto-CoT generates cluster centroids using Zero-Shot-CoT, but the rationales for these centroids may be flawed. In coevolutionary reasoning, examples iteratively serve as demonstrations for each other, refining and updating their rationales. Through this process, the accuracy of the demonstrations is gradually improved.

We address the first bias by introducing coevolutionary reasoning with judge to ensure iterative refinement progresses positively. To address the second bias, we balance "one demonstration for all" and "one demonstration for one" by randomly selecting examples from each cluster for iteration. Employing a judging process (e.g., ECHO+Judge) is expected to improve demonstration accuracy. However, due to judge bias—arising from potential errors in the judge's assessments—it does not always achieve this improvement (Figure 3). To alleviate judge bias, we propose P-sampling, where a sample is selected either from already chosen examples or new ones with probabilities P and $1-P$, respectively. This combination of P-sampling, global random sampling, and the judge mechanism addresses all three biases. We evaluate GCR on ten datasets across three reasoning problems. Experiments with GPT-3.5-turbo and GPT-4o-mini demonstrate that GCR consistently outperforms Auto-CoT on average while avoiding performance degradation caused by demonstration bias. Additionally, GCR is orthogonal to existing methods and can be seamlessly integrated with them.

2 Global Coevolutionary Reasoning

Overview The schematic illustration of our proposed approach is shown in Figure 4. GCR first uses Zero-Shot-CoT to generate answers for all N questions. Then, it applies K-means clustering to the (question, rationale) pairs, where the rationale derived from Zero-Shot-CoT serves as the basis for clustering. The intuition behind this approach is that we want the LLM to explore different reasoning paths when answering questions. By clustering based on the rationale, we aim to prevent reasoning errors from accumulating due to the lack of certain reasoning strategies. Next, a sample (question, rationale) is randomly selected from each cluster using P-sampling. These k samples are then subjected to T rounds of coevolutionary reasoning, where the question is re-answered, and the judge decides whether to update the rationale. This process, including both the sampling and the coevolutionary reasoning steps, is repeated until all samples have been re-answered. The total number of calls to the LLM in GCR is given by $N + \frac{N \times T}{1-P}$ (see Appendix C for the proof), and the intermediate process of coevolutionary reasoning is detailed in Appendix F.

2.1 Coevolutionary Reasoning

During the coevolutionary reasoning process, multiple samples serve as demonstrations for each other. Each question is iteratively re-answered, and the newly generated rationale is assessed to determine whether it should replace the existing rationale. If the new rationale demonstrates superior quality, it is updated; otherwise, the previous rationale is retained. The updated (question, rationale) pairs are then used as demonstrations for subsequent iterations of re-answering. The process of re-answering and updating all samples constitutes one iteration, and this procedure is repeated for T rounds to ensure convergence. The algorithm for coevolutionary reasoning is presented in Algorithm 1. See Appendix B for illustration of coevolutionary reasoning.

Algorithm 1 Coevolutionary Reasoning

Require: Question set $\mathcal{Q} = \{q_1, q_2, \ldots, q_k\}$, initial rationales $\mathcal{R} = \{r_1, r_2, \ldots, r_k\}$, number of iterations T
Ensure: Updated rationales $\mathcal{R}^*$, where $\mathcal{R}^*$ is the final updated set of rationales.

```
for t = 1 to T do                                          ▷ Iterate for T rounds
    for i = 1 to k do                                      ▷ Re-answer one by one
        Build demonstration set D_i = {(q_j, r_j) | j ≠ i}
        Generate new rationale: r'_i ← LLM(q_i | D_i)
        Evaluate new rationale: tag ← Judge(q_i, r'_i, r_i)
        if tag == True then                                ▷ If new rationale is better
            Update rationale: r_i ← r'_i
        end if
    end for
end for
return R*
```

2.2 Key Components of GCR

Judge The judge process evaluates whether a new rationale improves upon the original in the coevolutionary reasoning process. It serves two purposes:

- To eliminate the first type of bias, ensuring correct progression in iterations.
- To mitigate erroneous reasoning by reducing the impact of randomly sampling irrelevant demonstrations.

In this paper, we choose the judge method as follows:

1. **Self Judge:** Let the LLM judge whether the generated rationale is correct (prompting LLM to identify the specific reasons behind errors proves to be more difficult (Huang et al., 2024)). See Appendix A for the prompts. If

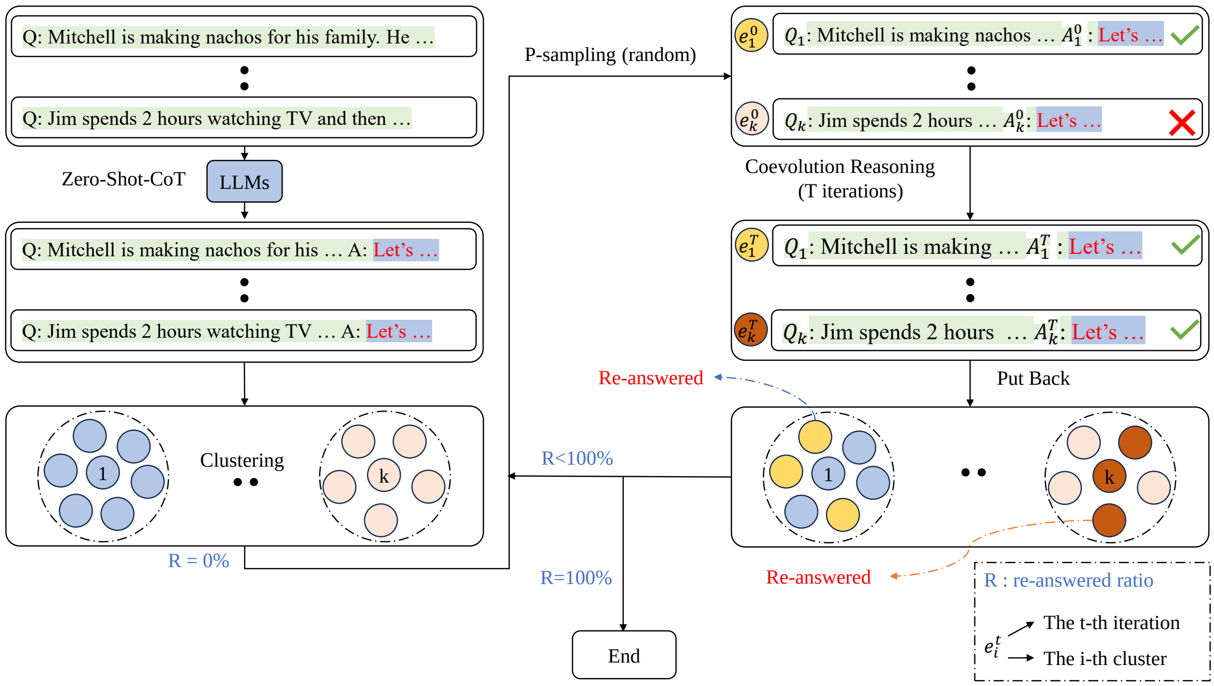

Figure 4: Illustrations of GCR. The process begins by generating initial answers for all N questions using Zero-Shot-CoT. Then, (question, rationale) pairs are encoded and clustered using K-means, where the rationale derived from Zero-Shot-CoT serves as the clustering basis. Next, a sample (question, rationale) is randomly selected from each cluster through P-sampling. These k selected samples undergo T rounds of coevolutionary reasoning, where the questions are re-answered and their rationales are iteratively refined. This process of sampling and coevolutionary reasoning continues until all samples have been re-answered, ensuring a more effective reasoning evolution.

the rationale is judged to be correct, the rationale is updated directly. Unless otherwise stated, the judge method used in this paper follows this approach for the sake of convenience. Note that the judge model should be identical to the generation model to ensure a fair comparison.

2. **Answer Entropy:** In line with the Self-Consistency (Wang et al., 2023b) idea, ask the large model to answer multiple times and use the entropy of the answers to judge. The smaller the entropy of the answers, the more accurate the demonstrations is considered. In this case, a majority voting mechanism is used to select the best answer and rationale, which are then updated.

3. **Probability Disparity:** Use the probability disparity (Wang and Zhou, 2024) of the generated rationale to judge (this needs to be done in the context of open-source models). If the disparity is higher than before, the rationale is updated.

4. **Oracle Labels:** Use ground truth to judge (it should be the performance upper bound for coevolutionary reasoning). If the answer corresponding to the generated rationale matches the ground truth, the rationale is updated.

From 1 to 4, the cost of the judge increases gradually, but so does the effectiveness of the judge (see Table 3).

Global random sampling Global random sampling differs from previous methods by not relying on a fixed set of demonstrations. Instead, for each question, demonstrations are randomly sampled from a global set, broadening the exploration of the prompt space. This approach has two main benefits:

- It eliminates the second type of bias by expanding the prompt space, preventing failures due to irrelevant demonstrations.
- It mitigates errors caused by judge bias, avoiding a significant performance drop in a "one prompt for all" scenario.

P-sampling Each time data is sampled for coevolutionary reasoning, there is a probability P of sampling from previously selected data and $1 - P$ from new data. P-sampling serves two purposes:

- It allows previously selected samples, which are assumed to be more accurate, to act as demonstrations, increasing the likelihood of correct answers for new samples.

- By enabling global random sampling, P-sampling lets samples re-enter the reasoning process with different demonstrations, expanding the exploration of the prompt space and improving the chance of correct answers.

Therefore, the three mechanisms work in synergy to perform reasoning and eliminate the three types of bias.

3 Experimental Setup

3.1 Benchmarks

Following Wang et al. (2023a) and Zhang et al. (2023), we evaluate GCR on ten benchmark datasets, categorized into three types of reasoning tasks: (i) arithmetic reasoning, comprising MultiArith (Roy and Roth, 2015), GSM8K (Cobbe et al., 2021), AddSub (Hosseini et al., 2014), AQUA-RAT (Ling et al., 2017), SingleEq (Koncel-Kedziorski et al., 2015), and SVAMP (Patel et al., 2021); (ii) commonsense reasoning, including CSQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021); and (iii) symbolic reasoning, with tasks like Last Letter Concatenation and CoinFlip (Wei et al., 2022). See statistical details in Table 5.

3.2 Baselines

We compare our proposed method, GCR, with three baseline methods: (1) **Zero-Shot-CoT.** Zero-Shot-CoT (Kojima et al., 2022) does not require any demonstrations, instead relying solely on a prompt to trigger the CoT reasoning. (2) **Auto-CoT.** Auto-CoT (Zhang et al., 2023) automatically selects examples by clustering with diversity and generates reasoning chains using Zero-Shot-CoT to construct demonstrations. (3) **ECHO.** ECHO (Jin and Lu, 2024) performs additional coevolutionary iterations on the demonstrations selected by Auto-CoT without any judgment.

3.3 Implementations

We conduct our experiments using GPT-3.5-turbo as the base language model. For validation, we also use GPT-4o-mini and mistral_7b_instruct_v3. To ensure the completeness of the response, we set a maximum CoT length of 1024 tokens, and for clustering, we apply Sentence-BERT with K-means. For a complete description of the experimental setup, including hyperparameters and additional implementation specifics, please refer to the Appendix D.

4 Experimental Results

The experimental results are displayed in Table 1. Overall, GCR outperforms Auto-CoT by a large margin and is more robust than ECHO. Across ten benchmark datasets, GCR achieves superior results with an average improvement of 6.1% and 2.1% over Auto-CoT on GPT-3.5-turbo and GPT-4o-mini, respectively. This demonstrates the effectiveness of our proposed approach in enhancing reasoning capabilities. However, as observed from Table 1, demonstration bias has a smaller impact on models with stronger reasoning capabilities. Additionally, due to the influence of judge bias, GCR exhibits a slight performance decline on certain datasets. In this section, we discuss the results of arithmetic reasoning, commonsense reasoning, and symbolic reasoning.

Arithmetic Reasoning: GCR achieves the best average performance compared with all baseline models, indicating the superiority of our method. Compared with the competitive baseline ECHO, GCR outperforms it by an average of 4.9% with GPT-3.5-turbo and 0.6% with GPT-4o-mini. The largest improvement is observed in GSM8K (+10.8%) and AQuA (+10.6%) under GPT-3.5-turbo, and in AQuA (+3.2%) under GPT-4o-mini. One possible reason is that these datasets suffer from two types of demonstration bias in the demonstrations automatically selected by Auto-CoT, namely, the selected demonstrations are not sufficiently representative or contain errors.

Commonsense Reasoning: Consistent improvement is observed in commonsense reasoning tasks. GCR outperforms all baseline models across both CSQA and StrategyQA. Compared with Auto-CoT, GCR improves the average performance by 7.1% under GPT-3.5-turbo and 0.5% under GPT-4o-mini, demonstrating its effectiveness in leveraging external knowledge and logical inference. These results suggest that GCR enhances the ability to reason about implicit knowledge and abstract relationships, making it more robust for real-world commonsense tasks.

Method	*Arithmetic*							*Commonsense*			*Symbolic*			Overall
	MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	avg.	CSQA	StrategyQA	avg.	LastLetter	Coin	avg.	
GPT-3.5-turbo														
Zero-Shot-CoT	94.2	71.7	81.3	54.3	89.4	73.9	77.5	74.1	60.8	67.5	56.6	71.0	63.8	72.7
Auto-CoT	92.8	72.0	80.5	59.4	89.4	74.8	78.2	74.0	59.7	66.8	82.6	92.2	87.4	77.7
ECHO	93.3	70.2	86.6	59.1	90.6	85.1	80.8	74.4	62.5	68.4	66.4	91.2	78.8	77.9
ECHO+Judge	90.7	67.9	87.1	58.3	88.8	**85.3**	79.7	57.4	**70.0**	63.7	**87.4**	**99.8**	**93.6**	79.3
GCR	**97.7**	**81.0**	**89.4**	**69.7**	**92.5**	83.9	**85.7**	**78.7**	69.2	**73.9**	84.6	91.0	87.8	**83.8**
GPT-4o-mini														
Zero-Shot-CoT	98.2	88.2	88.9	67.7	91.3	92.6	87.8	79.6	75.6	77.6	69.0	97.2	83.1	84.8
Auto-CoT	98.7	86.8	**90.4**	75.2	91.3	92.7	89.2	**83.1**	78.4	80.8	79.6	99.8	89.7	87.6
ECHO	**99.2**	90.4	**90.4**	79.1	91.7	92.9	90.6	82.9	79.0	81.0	88.4	**100.0**	**94.2**	89.4
ECHO+Judge	97.8	90.9	90.1	77.6	**92.3**	**94.0**	90.5	83.0	79.2	81.1	83.0	**100.0**	91.5	88.8
GCR	98.8	**91.0**	89.9	**82.3**	91.5	93.8	**91.2**	82.5	**80.1**	**81.3**	**88.6**	98.4	93.5	**89.7**

Table 1: Accuracy comparison across datasets using GPT-3.5-turbo and GPT-4o-mini, with avg representing average performance. **Bold** values indicate the best performance. ECHO+Judge incorporates a judging mechanism during iterative updates.

Symbolic Reasoning: GCR also achieves better results than Auto-CoT in symbolic reasoning tasks. Notably, due to the absence of a Judge mechanism to ensure the iterative process moves in the correct direction in ECHO (as shown in Figure 3), ECHO experiences a sharp performance drop on the LastLetter dataset in GPT-3.5-turbo compared to Auto-CoT (82.6 % $\rightarrow$ 66.4 %). However, GCR and ECHO+Judge, which incorporate the Judge process during iterations, do not exhibit such a performance decline. These results indicate that GCR addresses the demonstration bias present in Auto-CoT and exhibits strong robustness, avoiding sharp performance declines.

5 Analysis

Another Way to Judge To further validate the generalizability of GCR, we replaced the closed-source model's direct judgment-based method with the judge method designed for open-source models. We adopt the probability disparity proposed in Wang and Zhou (2024) to assess the quality of a rationale. The open-source model used is `mistral_7b_instruct_v3`. The probability disparity of a rationale is defined as:

$$\Delta_r = \frac{1}{|r|} \sum_{x_t \in r} \left(p(x_t^1 | x_{<t}) - p(x_t^2 | x_{<t}) \right) \qquad (1)$$

Here, r represents a rationale, and x_t^1 and x_t^2 are the top two tokens at the t-th decoding step, chosen based on their maximum post-softmax probabilities from the vocabulary, with x_t being part of the answer tokens. To avoid blind confidence, if the probability disparity for a token exceeds a threshold (in this section, 0.75), it is disregarded in the calculation. Table 2 presents a comparison of methods on different datasets. GCR achieves the best results across all datasets, further validating its effectiveness and generalizability.

Method	AQuA	GSM8K	StrategyQA	avg
Zero-Shot-CoT	29.9	45.8	63.1	46.3
Auto-CoT	36.6	55.7	68.0	53.4
ECHO	33.1	56.8	68.6	52.8
GCR	**39.4**	**57.4**	**69.7**	**55.5**

Table 2: Accuracy comparison of methods on different datasets using mistral_7b_instruct_v3 with probability disparity to judge in GCR. **Bold** values indicate the best performance.

Effect of Judge To verify the effectiveness of the Judge, we first incorporate the Judge process into ECHO's iterative rationale updating and observe how the accuracy of the demonstrations changes after iterations. As shown in Figure 3, ECHO+Judge helps mitigate the deviation from the correct direction during the iterative process. However, due to the influence of judge bias, there is a slight decrease in accuracy after the iterations. Additionally, we tested the reasoning performance of ECHO+Judge (see Table 1). Notably, using GPT-3.5-turbo on the LastLetter dataset, the demonstrations derived by ECHO+Judge exhibit higher accuracy compared to ECHO, avoiding the sharp performance drop observed in ECHO. However, on the CSQA dataset, ECHO+Judge also suffers a sharp performance decline. One possible reason for this is that, compared to the demonstrations derived by ECHO, those produced by ECHO+Judge, due to judge bias, have lower accuracy (see Figure 3). This also underscores the importance of considering the impact of judge bias.

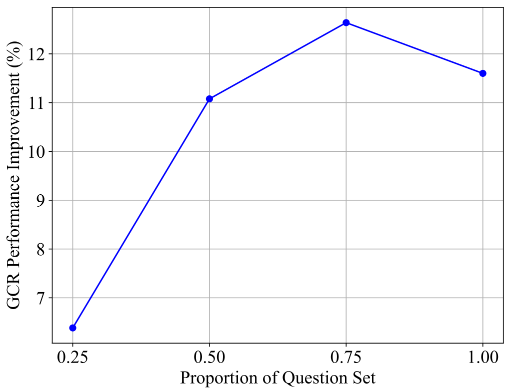

Figure 5: Impact of question set size on GCR's reasoning performance. The performance improvement of GCR over Zero-Shot CoT increases as the size of the question set grows. Larger question sets provide a broader exploration space for demonstrations, enhancing the likelihood of correct answers.

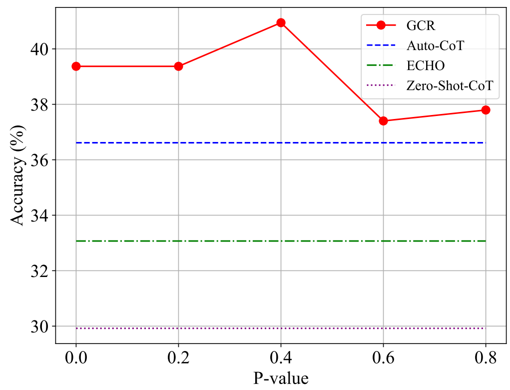

Figure 6: Comparison of GCR and baseline methods' reasoning performance at different P-values on the AQuA dataset using mistral_7b_instruct_v3.

Analysis of Global Random Sampling Compared to ECHO+Judge, GCR includes an additional global random sampling process, which allows us to assess whether this step contributes to improved reasoning performance. As shown in Table 1, GCR is more robust than ECHO+Judge. On the CSQA dataset with GPT-3.5-turbo, GCR benefits from the use of global random sampling, a decentralized approach that partially mitigates the impact of demonstration bias, thus preventing the sharp performance decline observed in ECHO+Judge. However, since global random sampling may select irrelevant or incorrect examples as demonstrations, and because judge bias cannot be completely eliminated, a slight decrease in reasoning performance is observed. Additionally, we investigated the impact of the question set size on GCR's reasoning performance using mistral_7b_instruct_v3 on the GSM8K dataset. We randomly sampled question sets of varying sizes from GSM8K and compared the performance improvement of GCR over Zero-Shot-CoT in reasoning. As shown in Figure 5, as the size of the question set increases, the reasoning performance of GCR improves more compared to Zero-Shot-CoT. This is because, with more questions, GCR has a larger exploration space for demonstrations due to global random sampling, making it more likely for the model to correctly answer the questions.

Analysis of P-sampling In this study, we investigate how to determine the optimal P-value in P-sampling and its impact on GCR's reasoning performance. We validated GCR's performance at different P-values using the AQuA dataset with the mistral_7b_instruct_v3 model, as shown in Figure 6. As the P-value increases, the probability of sampling previously selected examples during coevolutionary reasoning also rises. This suggests that GCR's reasoning performance may improve, but at the cost of increased computational overhead. However, as observed in Figure 6, when the P-value is between 0.2 and 0.4, GCR's reasoning performance is highest. When the P-value reaches 0.6, 0.8, or higher, performance does not continue to increase or converge as expected. Therefore, we recommend setting the P-value between 0.2 and 0.4 to achieve a balance between computational efficiency and high reasoning performance.

Effect of Coevolutionary Reasoning This section explores the role of coevolutionary reasoning, which involves multiple answers to the same question, similar to Self-Consistency (Wang et al., 2023b), Multiagent Debate (Du et al., 2024), and multiple calls to LLM with a judge. As suggested by Huang et al. (2024), when comparing reasoning performance, it is important to ensure consistent overhead. We first investigate the performance comparison between GCR and Auto-CoT with multiple calls to LLM under two conditions: using oracle labels for judgment and allowing the large model to judge itself, both using GPT-3.5-turbo (see Appendix E for experimental details). From Table 3, we observe that under oracle label judgment, GCR and multi-call achieve similar performance, but GCR requires significantly less over-

Method	Judge method	GSM8K	AQuA	SVAMP
Auto-CoT+multi-call ($T_{\max} = 10$)	Oracle Labels	-	93.3	-
GCR ($T_{\max} = 4$)	Oracle Labels	-	92.1	-
GCR ($T_{\max} = 6$)	Oracle Labels	-	93.7	-
Auto-CoT+multi-call ($T = 5$)	Self Judge	71.9	67.3	79.9
GCR ($T = 4$)	Self Judge	81.0	69.7	83.9
Auto-CoT+Self-Consistency ($T = 50$)	N/A	-	70.5	-
GCR ($T = 4 \times 10$)	Answer Entropy	-	72.0	-

Table 3: Accuracy comparison of GCR with other methods that also require multiple answers to the same question, using GPT-3.5-turbo. The '-' in the table indicates that no experiments were conducted due to overhead.

head. When using Self Judge, GCR outperforms multi-call while maintaining the same overhead. Moreover, GCR and Self-Consistency are orthogonal, and when using Answer Entropy for judgment, GCR combined with Self-Consistency achieves better performance than Auto-CoT combined with Self-Consistency under the same overhead. Therefore, coevolutionary reasoning can achieve comparable performance to multi-call LLM with much less overhead. Multiagent Debate involves multiple agents engaging in a multi-round debate on the same question, whereas GCR involves coevolutionary reasoning across multiple questions.

6 Related Work

Chain of Thought Prompting Chain-of-Thought (CoT) reasoning has been widely explored to enhance LLM inference. Wei et al. (2022) first introduced CoT prompting in a few-shot setting, while Kojima et al. (2022) extended it to the zero-shot scenario. Auto-CoT (Zhang et al., 2023) automatically constructs demonstrations, achieving performance comparable to Few-Shot-CoT in a zero-shot setting. Active Prompting (Diao et al., 2024) further improves prompting by annotating the most important task-specific questions as demonstrations.

Judge To enhance answer verification, Weng et al. (2023) treat the conclusion obtained by CoT as a condition for solving the original problem, performing backward verification. Madaan et al. (2024) propose an iterative self-refinement process where an LLM generates an initial output, critiques its own response, and revises it accordingly. Ling et al. (2024) introduce the Natural Program format for deductive reasoning, facilitating the structured extraction and verification of reasoning steps. However, Huang et al. (2024) argue that LLMs still struggle with self-correction in the absence of external feedback. Several works address uncertainty estimation in reasoning. Wan et al. (2023) leverage answer entropy to measure uncertainty and highlight the importance of diverse demonstrations, noting that Auto-CoT, which selects demonstrations based solely on question embeddings, lacks control over rationale quality and may produce misleading demonstrations. Li et al. (2022) improve reasoning by generating diverse prompts to explore multiple reasoning paths, employing a trained verifier with weighted voting to filter incorrect answers, and verifying each reasoning step individually. Additionally, Kuhn et al. (2023) introduce semantic entropy to quantify uncertainty in natural language generation, while Wang and Zhou (2024) use probability disparity to assess rationale accuracy in open-source settings.

Evolve in LLMs Guo et al. (2024) propose an approach where multiple initial prompts are generated, with LLMs acting as evolutionary operators to iteratively refine them based on their performance on a development set. Similarly, Jin et al. (2024) introduce an evolutionary algorithm where CoT prompts undergo crossover, mutation, and rewriting to enhance problem understanding. However, these methods focus solely on prompt evolution, neglecting demonstration refinement. Xu et al. (2024a) propose Reprompting, which iteratively expands an initial set of zero-shot-generated recipes by using previous samples as parent prompts, filtering out ineffective ones based on answer correctness. While Reprompting can be seen as evolving demonstrations, it relies on ground-truth answers for evaluation. In contrast, GCR evolves demonstrations independently of answer supervision, offering a distinct approach to demonstration refinement.

7 Conclusion

This paper presents a method for global coevolutionary reasoning, referred to as GCR, where samples are clustered based on rationales obtained through Zero-Shot-CoT. Within each cluster, samples are randomly selected using P-sampling, and coevolutionary iterations are performed where samples act as demonstrations for one another. This approach addresses the demonstration bias inherent in Auto-CoT, enhancing reasoning performance in the absence of manually designed demonstrations. Furthermore, it can be integrated with existing methods for reasoning tasks.

Limitations

While our approach demonstrates effectiveness in improving reasoning performance, it has several limitations:

1. **Computational Cost**: The iterative process in coevolutionary reasoning increases the number of LLM calls, leading to a significant increase in computational overhead, although it maintains less overhead compared to Self-Consistency.

2. **Dependence on Judge Quality**: The effectiveness of the coevolutionary process relies on the quality of the judge. Although GCR mitigates judge bias to some extent through global sampling and P-sampling, a stronger judge could further enhance reasoning performance. Future work will explore better methods for evaluating the quality of rationales in both open-source and closed-source settings.

3. **Assumption of a Fixed Dataset**: Most CoT studies assume access to a complete dataset with test questions (Wei et al., 2022; Kojima et al., 2022). Future work could extend GCR to a streaming setting where data arrives dynamically.

4. **Understanding LLM Limitations**: It remains an open question whether LLMs lack the capability to perform a certain class of reasoning methods or if they struggle with specific problem types. Further research is needed to disentangle these factors.

Ethics Statement

In this work, we use publicly available benchmarks under their respective licenses. GSM8K and SVAMP use the MIT License, AQUA and StrategyQA use the Apache-2.0 License, and the remaining datasets are unspecified. These publicly available datasets are checked to ensure that they do not contain any offensive or illegal content.

References

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. 2023. Skill-based few-shot selection for in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13472–13492.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1350, Bangkok, Thailand. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Feihu Jin, Yifan Liu, and Ying Tan. 2024. Zero-shot chain-of-thought reasoning guided by evolutionary algorithms in large language models. *arXiv preprint arXiv:2402.05376*.

Ziqi Jin and Wei Lu. 2024. Self-harmonized chain of thought. *arXiv preprint arXiv:2409.04057*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022. Making large language models better reasoners with step-aware verifier. *arXiv preprint arXiv:2206.02336*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2024. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752.

Joseph Shtok, Amit Alfassy, Foad Abo Dahood, Eliyahu Schwartz, Sivan Doveh, and Assaf Arbelle. 2024. Augmenting in-context-learning in llms via automatic data labeling and refinement. *arXiv preprint arXiv:2410.10348*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

Weijia Xu, Andrzej Banburski, and Nebojsa Jojic. 2024a. Reprompting: Automated chain-of-thought prompt inference through gibbs sampling. In *Forty-first International Conference on Machine Learning*.

Zifan Xu, Haozhu Wang, Dmitriy Bespalov, Xian Wu, Peter Stone, and Yanjun Qi. 2024b. Lars: Latent reasoning skills for chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3624–3643.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Xiaosong Yuan, Chen Shen, Shaotian Yan, Xiao Feng Zhang, Liang Xie, Wenxiao Wang, Renchu Guan, Ying Wang, and Jieping Ye. 2024. Instance-adaptive zero-shot chain-of-thought prompting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xiang Zhang and Dujian Ding. 2024. Supervised chain of thought. *arXiv preprint arXiv:2410.14198*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

A Prompts for Judging

We use task-specific judgment by querying the LLM. The specific prompts are shown in Table 4. Although we provide the LLM with a prompt to give specific reasons when determining correctness, we only use the conclusion it provides. This is because we found that, compared to directly providing the conclusion, asking for specific reasons leads to more accurate judgments.

B Illustration of Coevolutionary Reasoning

The illustration of coevolutionary reasoning is shown in Figure 7.

C Proof of the Expected Number of LLM Calls in GCR

We first consider a scenario where there is a single cluster of data, with the cluster containing n data points. In each sampling step, there is a probability P of selecting a sample from the already sampled data, and a probability $1 - P$ of selecting a sample from the data that has not been sampled yet. The expected number of steps required to sample all n data points in this cluster is:

$$E = \sum_{k=n}^{\infty} k \binom{k-1}{n-1} (1-P)^{n-1} P^{k-n} (1-P)$$

Let $t = k - n$, then we have:

$$\begin{aligned} E &= \sum_{t=0}^{\infty} (n+t) \binom{t+n-1}{n-1} (1-P)^n P^t \\ &= (1-P)^n \left[n \sum_{t=0}^{\infty} P^t \binom{t+n-1}{n-1} \right. \\ &\quad \left. + \sum_{t=0}^{\infty} t P^t \binom{t+n-1}{n-1} \right] \\ &= (1-P)^n \left[n \sum_{t=0}^{\infty} P^t \binom{t+n-1}{t} \right. \\ &\quad \left. + \sum_{t=0}^{\infty} t P^t \binom{t+n-1}{t} \right] \end{aligned} \tag{2}$$

Next, we prove the following identity:

$$\sum_{m=0}^{\infty} \binom{m+k-1}{m} x^m = \frac{1}{(1-x)^k}, \qquad 0 \le x < 1, k \ge 1 \tag{3}$$

When $k = 1$,

$$\sum_{m=0}^{\infty} \binom{m+k-1}{m} x^m = \sum_{m=0}^{\infty} x^m = \frac{1}{1-x}. \tag{4}$$

This equation holds. When $k \ge 2$, we hypothesize that the equation holds for $k - 1$, then

$$\sum_{m=0}^{\infty} \binom{m+k-2}{m} x^m = \frac{1}{(1-x)^{k-1}}. \tag{5}$$

Thus,

$$\begin{aligned} \sum_{m=0}^{\infty} \binom{m+k-1}{m} x^m &= \sum_{m=0}^{\infty} \binom{m+k-2}{m} x^m \\ &\quad + \sum_{m=1}^{\infty} \binom{m+k-2}{m-1} x^m. \end{aligned} \tag{6}$$

Now, we can express the second sum as:

$$\sum_{m=1}^{\infty} \binom{m+k-2}{m-1} x^m = x \sum_{m=0}^{\infty} \binom{m+k-1}{m} x^m. \tag{7}$$

Let $T = \sum_{m=0}^{\infty} \binom{m+k-1}{m} x^m$. Then, we have:

$$T = \frac{1}{(1-x)^{k-1}} + xT,$$

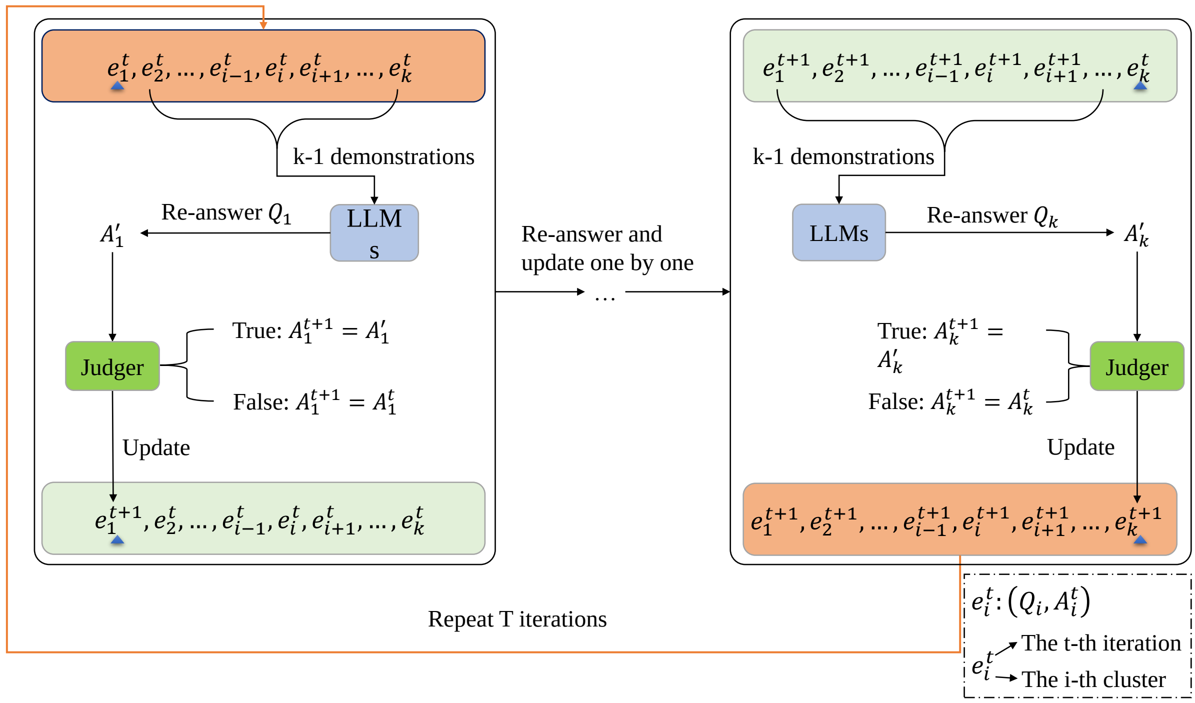

Figure 7: Illustration of coevolutionary reasoning.

T = 0

Q: Mitchell is making nachos for his family. He buys two bags of chips with 55 chips each. If his family has five members, how many chips does each person get if they all get the same number?
A:|1|How many chips does Mitchell have in total?|2 bags of chips x 55 chips each|110 chips| |2|How many chips does each family member get?|Total chips / number of family members|110 chips / 5 members|22 chips| Therefore, each family member gets 22 chips.

:

Q: Jim spends 2 hours watching TV and then decides to go to bed and reads for half as long. He does this 3 times a week. How many hours does he spend on TV and reading in 4 weeks?
A:| 1 | How many hours does Jim spend watching TV and reading each week? | 2 hours of watching TV + 1 hour of reading = 3 hours | | 2 | How many hours does Jim spend on TV and reading in 4 weeks? | 3 hours/week * 4 weeks = 12 hours | Jim spends a total of 12 hours on TV and reading in 4 weeks.

:

T = 1

Q: Mitchell is making nachos for his family. He buys two bags of chips with 55 chips each. If his family has five members, how many chips does each person get if they all get the same number?
A:Mitchell buys 2 bags of chips, each with 55 chips, so he has a total of 2 * 55 = 110 chips. Mitchell's family has 5 members. To find out how many chips each person gets, we need to divide the total number of chips by the number of family members: 110 chips / 5 people = 22 chips per person. Therefore, each person in Mitchell's family will get 22 chips.

:

Q: Jim spends 2 hours watching TV and then decides to go to bed and reads for half as long. He does this 3 times a week. How many hours does he spend on TV and reading in 4 weeks?
A:Jim watches TV for 2 hours and then reads for 2 / 2 = 1 hour. So, in one day, Jim spends a total of 2 + 1 = 3 hours on TV and reading.\n-Since he does this 3 times a week, in one week he spends 3 hours/day x 3 days/week = 9 hours on TV and reading. In 4 weeks, he will spend a total of 9 hours/week x 4 weeks = 36 hours on TV and reading. Therefore, Jim spends 36 hours on TV and reading in 4 weeks.

:

T = 2

Q: Mitchell is making nachos for his family. He buys two bags of chips with 55 chips each. If his family has five members, how many chips does each person get if they all get the same number?
A:|1|How many chips does Mitchell have in total?|2 bags of chips x 55 chips each|110 chips| |2|How many chips does each family member get?|Total chips / number of family members|110 chips / 5 members|22 chips| Therefore, each family member gets 22 chips.

:

Q: Jim spends 2 hours watching TV and then decides to go to bed and reads for half as long. He does this 3 times a week. How many hours does he spend on TV and reading in 4 weeks?
A:| 1 | How many hours does Jim spend watching TV and reading each week? | 2 hours of watching TV + 1 hour of reading = 3 hours | | 2 | How many hours does Jim spend on TV and reading in 4 weeks? | 3 hours/week * 4 weeks = 12 hours | Jim spends a total of 12 hours on TV and reading in 4 weeks.

:

Figure 8: An example of intermediate iteration process in coevolutionary reasoning.

Task	Prompt
Arithmetic Reasoning	Input = [instruction, original problem, solution] Output = [yes or no, reason] Instruction: Is this solution accurate in terms of calculation errors, missing-step errors, and semantic misunderstanding errors? Please answer "yes" or "no" and provide a reason. Carefully check: 1. Calculation errors: Are there any arithmetic or algebraic errors in the steps or final result? 2. Missing-step errors: Are there any steps omitted that are necessary for correctly solving the problem? 3. Semantic misunderstanding errors: Does either solution misunderstand the problem or apply incorrect methods or formulas? original problem: <ORG_PROB> solution: <ORG_SOL>
Commonsense Reasoning	Input = [instruction, original problem, solution] Output = [yes or no, reason] Instruction: From a common-sense and logical reasoning perspective, is this solution accurate? Please answer "yes" or "no", and follow the given output format without any additional information. original problem: <ORG_PROB> solution: <ORG_SOL>
Symbolic Reasoning	Input = [instruction, original problem, solution] Output = [yes or no, reason] Instruction: From a symbolic and logical reasoning perspective, is this solution accurate? Please answer "yes" or "no", and follow the given output format without any additional information. original problem: <ORG_PROB> solution: <ORG_SOL>

Table 4: Prompts for different reasoning tasks to evaluate rationales.

which simplifies to:

$$T = \frac{1}{(1-x)^k}.$$

Thus, the equation 3 holds for k.

By mathematical induction, we conclude that the equation 3 holds for all $k \geq 1$.

From equation 3 and equation 2, we obtain:

$$\begin{aligned} E &= (1-P)^n \left[\frac{n}{(1-P)^n} + P \cdot \frac{d}{dP} \left(\frac{1}{(1-P)^n} \right) \right] \\ &= (1-P)^n \left[\frac{n}{(1-P)^n} + \frac{Pn}{(1-P)^{n+1}} \right] \\ &= n + n\frac{P}{1-P} \\ &= \frac{n}{1-P}. \end{aligned} \tag{8}$$

Thus, in a single cluster, the expectation of P-sampling is $\frac{n}{1-P}$.

Now, assuming the clustering is uniform, the expected total number of LLM calls is:

$$E_{\text{total}} = \frac{\frac{N}{C} \times T}{1-P} \times C + N = \frac{NT}{1-P} + N,$$

where C is the number of clusters, the term N represents the initial number of Zero-Shot-CoT calls, and the term $\frac{NT}{1-P}$ accounts for the LLM calls during the P-sampling and coevolutionary reasoning process.

D Experimental Details on the main experiments

We mainly use GPT-3.5-turbo as language model.[1] Furthermore, to validate the generalizability of our method, we also conducted experiments on GPT-4o-mini and mistral_7b_instruct_v3. Except for Zero-Shot-CoT, which uses the CoT trigger `"\n\n|step|subquestion|process|result|"`, all other methods use the CoT trigger `"Let's think step by step"`. For Auto-CoT and ECHO, we use Sentence-BERT (Reimers and Gurevych, 2019) to encode questions and apply K-means for clustering following Zhang et al. (2023), and the number of demonstrations k is 8 except for AQuA and LastLetter(4), CSQA(7), and StrategyQA(6) following Wei et al. (2022). Here, the number of demonstrations k is equivalent to the number of clusters. Due to the observation that, in the original Auto-CoT paper setup, with a maximum CoT length of 256, the rationale was sometimes incomplete, we set the maximum CoT length to 1024 in all our experiments to ensure the generated rationale is complete. In experiments with GPT-4o-mini and mistral_7b_instruct_v3, demonstrations generated using the original Auto-CoT method (selecting the examples closest to the cluster center with fewer than 6 reasoning steps) sometimes did not meet this condition. Therefore, in the experiments with GPT-4o-mini and mistral_7b_instruct_v3, we removed the restriction of fewer than 6 reasoning steps to obtain the demonstrations. In ECHO, the number of iterations for demonstrations is set to 4. For GCR, we use Sentence-BERT to encode the rationale and apply K-means for clustering. In P-sampling, P is set to 0.2, and the number of coevolutionary reasoning iterations is set to 4. To ensure a fair comparison, the number of clusters is set to 5, so that during the coevolutionary reasoning process, the number of demonstrations when re-answering a question is 4 (which matches the minimum number of demonstrations in Auto-CoT or ECHO). Unless otherwise specified, the temperature is set to 0 for experimental reproducibility.

[1]We conducted the experiments using this model between October 2024 and November 2024.

E Experimental Details on the Effect of Coevolutionary Reasoning

We investigate the role of coevolutionary reasoning using GPT-3.5-turbo. In the case of using Oracle Labels for judgment, we compare GCR ($T_{max} = 4$), GCR ($T_{max} = 6$) with Auto-CoT+multi-call. Here, T_{max} refers to the maximum number of coevolutionary reasoning iterations, which is set to 4 or 6, and the iteration stops once the answer is correct. Auto-CoT+multi-call refers to using Auto-CoT to construct a demonstration set, followed by multiple answers. After each answer, Oracle Labels are used to check if the answer is correct, and if so, the next question is answered. For Auto-CoT+multi-call, the maximum number of answers, T'_{max}, is set to 10. To ensure nearly identical computational costs, we require that $\frac{T_{max}}{1-P}$ must be less than or equal to T'_{max} (where P is set to 0.2 by default), meaning T_{max} should be less than or equal to $T'_{max} \times (1-P) = 8$.

In the case of using Self-Judge for judgment, we again ensure nearly identical computational costs and compare GCR ($T = 4$) with Auto-CoT+multi-call ($T = 5$). In this process, the LLM itself decides whether to update the answer after each response.

To compare with Auto-CoT+Self-Consistency, we use Answer Entropy for judgment in GCR. Coevolutionary reasoning iterates for 4 steps, and after each question, the LLM answers 10 times to generate multiple different answers (to ensure diversity, the temperature for GCR is set to 0.7). The entropy is calculated, and if the entropy is smaller than the previous iteration, the answer and rationale are updated with the majority-voted answer and corresponding rationale (the answer may have multiple rationales, and the shortest one is chosen). Otherwise, no update is made. Auto-CoT+Self-Consistency answers 50 times during question answering $(4 \times 10/(1 - P))$.

Following Wang et al. (2023b), both Auto-CoT+multi-call and Auto-CoT+Self-Consistency are set with a temperature of 0.7. Unless otherwise specified, the temperature for GCR is set to 0.

F Coevolutionary Example Iteration Process

The iteration of demonstrations in ECHO+Judge can be seen as a process of coevolutionary reasoning. Therefore, in this section, we use GPT-3.5-turbo to present the intermediate process of constructing demonstrations in ECHO+Judge for GSM8K as an example of coevolutionary reasoning. As shown in Figure 8, after one round of coevolutionary reasoning, the questions are answered correctly. However, due to the influence of judge bias, the second iteration results in an incorrect answer for a certain question, highlighting the importance of mitigating judge bias.

G Details of the Datasets Evaluated

The details of datasets being evaluated are shown in Table 5.

Dataset	Domain	# Samples	Ave. words	Answer
MultiArith	Math	600	31.8	Number
AddSub	Math	395	31.5	Number
GSM8K	Math	1319	46.9	Number
AQUA	Math	254	51.9	Option
SingleEq	Math	508	27.4	Number
SVAMP	Math	1000	31.8	Number
CSQA	CS	1221	27.8	Option
StrategyQA	CS	2290	9.6	Yes / No
Last Letters	Sym.	500	15.0	String
Coin Flip	Sym.	500	37.0	Yes / No

Table 5: Details of datasets being evaluated. Math: arithmetic reasoning. CS: commonsense reasoning. Sym.: symbolic reasoning.