

A Appendix

A.1 Theoretical Analysis of Eq. 4

In Eq. 4, we use \mathcal{S} sampled from the language model $\text{LM}_{\mathcal{G}}$ generations. This is an approximation for sampling from the infinite set of all possible chaining paths. And the finite set \mathcal{S} is constructed based on the generation $\text{LM } P_{\text{LM}_{\mathcal{G}}}$, which is different from our target distribution as shown in Eq. 4.

Specifically, denote the infinite set of all possible generated completions till the t -th step as \mathcal{S}^* , we approximate sampling from $P_{beam}^*(s^{1:t}) = \frac{\exp(\mathcal{E}(s^{1:t})/\tau)}{\sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau)}$ via $P_{beam}(s^{1:t}) = \frac{\exp(\mathcal{E}(s^{1:t})/\tau)}{\sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau)}$, where \mathcal{S} is the approximation of \mathcal{S}^* with $|\mathcal{S}| = nk = M \leq |\mathcal{S}^*|$.

Define the upper bound \bar{c} and the lower bound \underline{c} on each $\exp(\mathcal{E}(s^{1:t})/\tau)$ as $\bar{c} \geq \exp(\mathcal{E}(s^{1:t})/\tau) \geq \underline{c}$ for all $s^{1:t} \in \mathcal{S}^*$. Define the ratio as $r = \bar{c}/\underline{c}$. Note that $\underline{c} \geq 1$ since $\mathcal{E}(s^{1:t})/\tau \geq 0$. Thus, we can take $r \leq \bar{c}$.

We now give the following proposition which shows that $|P_{beam}^*(s^{1:t}) - P_{beam}(s^{1:t})|$ decreases at the rate of $\mathcal{O}(\frac{1-M/|\mathcal{S}^*|}{M})$ toward 0 as M increases. Note that as M increases toward $|\mathcal{S}^*|$, the numerator $1 - M/|\mathcal{S}^*|$ decreases toward 0 while the factor for the denominator $\frac{1}{M}$ also decreases.

Proposition 1. For any $s^{1:t}$, the difference between $P_{beam}^*(s^{1:t})$ and $P_{beam}(s^{1:t})$ is bounded by

$$|P_{beam}^*(s^{1:t}) - P_{beam}(s^{1:t})| \leq r^2 \left(\frac{1 - M/|\mathcal{S}^*|}{M} \right) \quad (6)$$

Proof. We now prove the second statement by analyzing the absolute difference:

$$|P_{beam}^*(s^{1:t}) - P_{beam}(s^{1:t})| \quad (7)$$

$$= \left| \frac{\exp(\mathcal{E}(s^{1:t})/\tau)}{\sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau)} - \frac{\exp(\mathcal{E}(s^{1:t})/\tau)}{\sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau)} \right| \quad (8)$$

$$= \frac{\exp(\mathcal{E}(s^{1:t})/\tau) \left| \sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau) - \sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau) \right|}{\left(\sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau) \sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau) \right)} \quad (9)$$

$$= \frac{\exp(\mathcal{E}(s^{1:t})/\tau) \left| \sum_{s^{1:t} \in \mathcal{S}^* \setminus \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau) \right|}{\left(\sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau) \right) \sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau)} \quad (10)$$

Since $\exp(\mathcal{E}(s^{1:t})/\tau)$ is nonnegative, using the upper bound on each $\exp(\mathcal{E}(s^{1:t})/\tau)$, we have:

$$|P_{beam}^*(s^{1:t}) - P_{beam}(s^{1:t})| \leq \frac{\bar{c}^2(|\mathcal{S}^*| - M)}{\left(\sum_{s^{1:t} \in \mathcal{S}} \exp(\mathcal{E}(s^{1:t})/\tau) \right) \sum_{s^{1:t} \in \mathcal{S}^*} \exp(\mathcal{E}(s^{1:t})/\tau)} \quad (11)$$

Similarly, using the lower bound on each $\exp(\mathcal{E}(s^{1:t})/\tau)$,

$$|P_{beam}^*(s^{1:t}) - P_{beam}(s^{1:t})| \leq \frac{\bar{c}^2(|\mathcal{S}^*| - M)}{\underline{c}^2|\mathcal{S}^*|M} = r^2 \left(\frac{1 - M/|\mathcal{S}^*|}{M} \right) \quad (12)$$

447

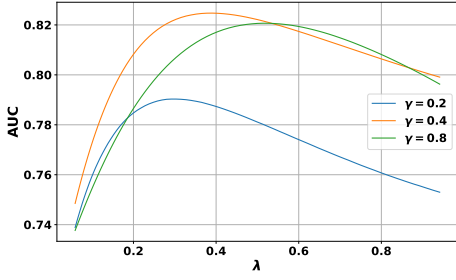
□

A.2 Extended Experiments

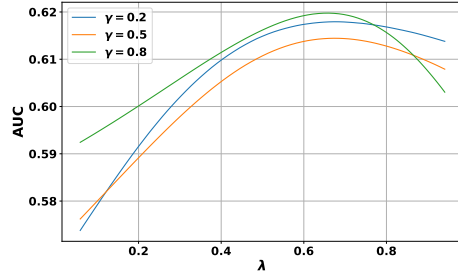
More Analysis on Self-Evaluation Recall that we use a combination of generation confidence and faithfulness score as $\mathcal{E}_{\lambda} = \mathcal{C}^{\lambda} \cdot \mathcal{P}^{(1-\lambda)}$, with $\lambda \in [0, 1]$. In our experiments, we set $\lambda = 0.5$ for all tasks for simplicity. However, we investigate its effects here since, intuitively, it is an important hyperparameter for distinguishing correct / incorrect predictions and might require different values for various reasoning tasks and datasets. Its effect is also coupled with the language model temperature γ .

Figure 6 demonstrates how λ functions on arithmetic (GSM8K) and commonsense (StrategyQA). In general, we observe that the performance remains relatively stable with different choices of λ on different datasets, although fine-tuning this hyperparameter might lead to further improvements. This stability suggests that the choice of λ is not overly sensitive across various reasoning tasks and datasets, but exploring its optimal value for specific tasks could potentially lead to even better performances.

To examine the influence of incorporating faithfulness on LLM final predictions, we plot the distributions of changes in different scores, specifically the faithfulness score \mathcal{C} , the generation confidence \mathcal{P} , and the overall decoding score \mathcal{E}_{λ} on the baseline reasoning chains and the reasoning chains generated by our method. We categorize the data points into 4 sets based on whether our approach changes the final prediction. Since the

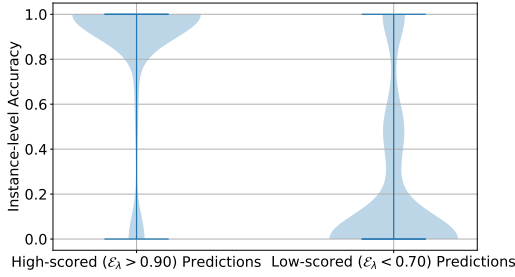


(a) λ -AUC curves of \mathcal{E}_λ on GSM8K (PAL).

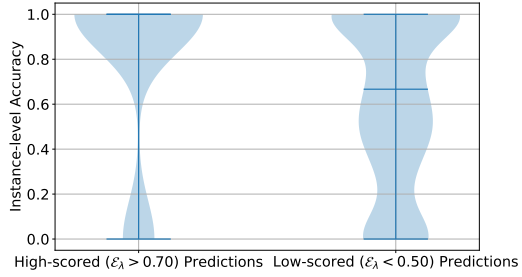


(b) λ -AUC curves of \mathcal{E}_λ on StrategyQA (CoT).

Figure 6: The change of AUC scores with different values of λ in \mathcal{E}_λ . We calculate the AUC score as how \mathcal{E}_λ can successfully determine whether the corresponding predicted reasoning chain can produce the ground-truth answer. The predictions here are from the baseline methods (*i.e.*, CoT & PAL) with different LM temperatures γ , as represented by curves of different colors.



(a) GSM8K (PAL prompting)



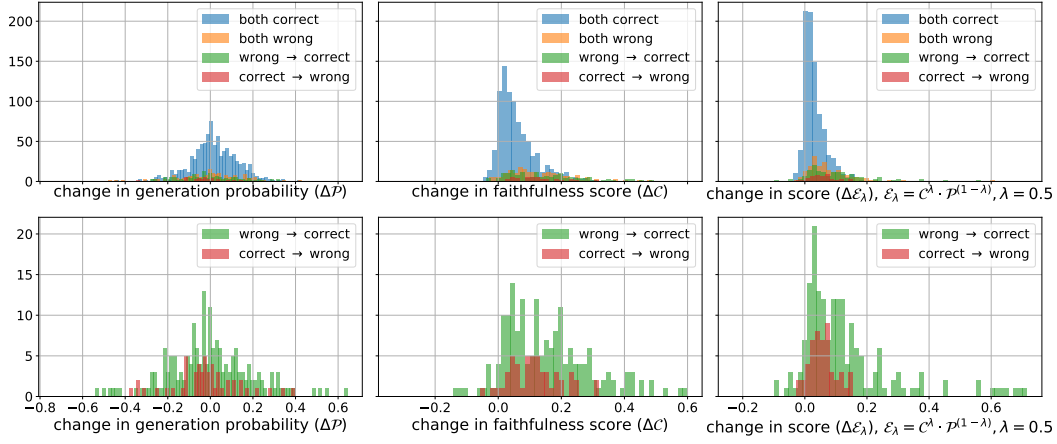
(b) StrategyQA (CoT prompting)

Figure 7: Comparison between predictions of high v.s. low self-evaluation scores on instance-level accuracy.

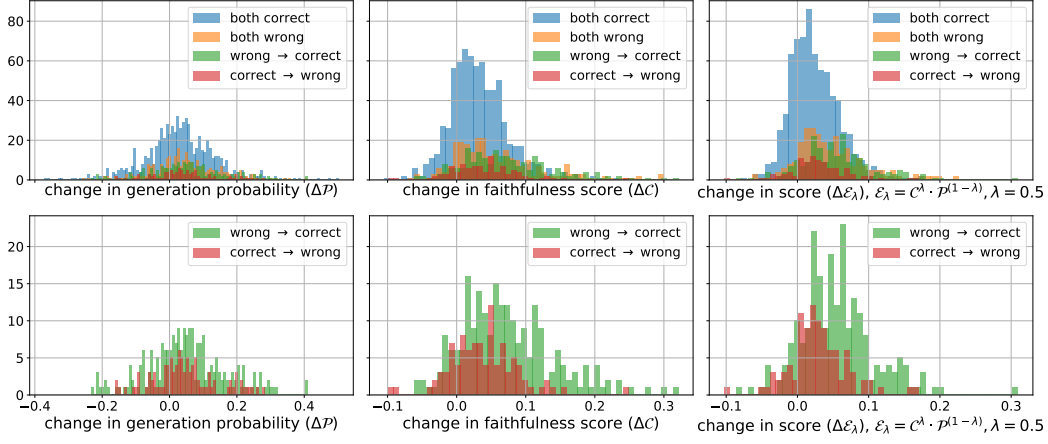
majority of the data points belong to the “both correct” set (in blue), where both baselines and our method generate accurate predictions, we particularly highlight the last two sets (in green and red), where our method results in improvement and degradation, respectively.

As shown in Figure 8, faithfulness typically works by significantly increasing the evaluation confidence \mathcal{C} of model predictions, while the generation confidence \mathcal{P} remains similar to that of the baseline methods. Specifically, for the evaluation confidence \mathcal{C} , our approach corrects the original predictions by increasing the confidence scores. This indicates that evaluation confidence plays a crucial role in guiding the decoding toward a better reasoning choice in decomposed reasoning. The increase is more significant for PAL when compared with CoT. This demonstrates that LLMs are generally better at judging the logic in reasoning that is more structured, while the free-text intermediate steps (*e.g.*, CoT reasoning) may be challenging to conduct information extraction and soundness checking.

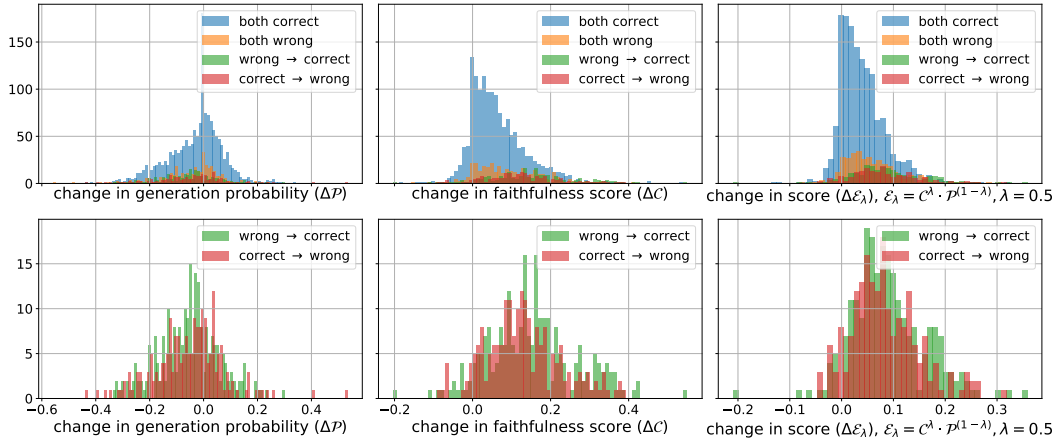
A similar conclusion can be drawn from Figure 7, where the difference in instance-level accuracy distributions between high-scored and low-scored predictions is more significant on the GSM8K dataset. For StrategyQA, while the incorporation of faithfulness helps, the level of the score value does not align well with whether the prediction is correct. For example, most of the low-scored predictions can still obtain the correct answers, as shown by the plot on the right of Figure 7b.



(a) Distributions of score shifts on GSM8K using PAL prompting.



(b) Distributions of score shifts on GSM8K using CoT prompting.



(c) Distributions of score shifts on StrategyQA using CoT prompting.

Figure 8: Distributions of changes in scores from baselines to our method. Since the prediction correctness keeps unchanged most of the time (*i.e.*, “both correct/incorrect” in blue/orange), we specifically plot how the scores shift on data points where the predictions get corrected or incorrect, as shown in green and red, respectively.

GPT-4 Experiments The recently launched GPT-4 has demonstrated notable improvements in reasoning capabilities across a variety of tasks. In this section, we examine and compare the reasoning skills of different large language models (LLMs), specifically Codex and GPT-4, in assessing and determining the accuracy of each step in a reasoning chain. We contrast the confidence scores and corresponding explanations for Codex (\mathcal{C}) and GPT-4 (\mathcal{S}) in the context of both arithmetic and commonsense reasoning, as shown in Figure 10 and Figure 11, respectively. For ease of visualization, we employ the same colormap (shown in Figure 9) as in Figure 5 to represent the scale of scores. Since OpenAI has not provided access to the token-wise likelihood of generated text, we directly request GPT-4 to score the reasoning steps using binary values⁵. Moreover, we report the average of three evaluation results to reduce the variance of sampling discrete values, i.e., $S = (S_1 + S_2 + S_3)/3, S_i \in [0, 1]$.

As illustrated in Figure 10, GPT-4 demonstrates greater effectiveness in pinpointing the central logical error in arithmetic reasoning. For instance, we can observe that $S < \mathcal{C}$ for `alex_total = alex_weight + weight_multiplier * grace_weight` and $S > \mathcal{C}$ for `answer = grace_weight + alex_total`, where the former leads to an incorrect final answer. Additionally, GPT-4 typically offers detailed explanations and alternative solutions. As seen in the step `answer = grace_weight + alex_total`, GPT-4 can correct minor errors even when it arrives at the correct final answer. However, GPT-4 may still encounter difficulties in detecting small errors within the text, which can have a significant impact on logical consistency. This challenge is illustrated by the substantial variance in \mathcal{S} for the step `alex_total = alex_weight + weight_multiplier * grace_weight`.

The benefits of well-crafted explanations in GPT-4 become more significant when handling complex reasoning tasks, as demonstrated in Figure 11. For instance, in the R_{42} of Q_4 shown in Figure 5b, Codex generally assigns high evaluation scores for all steps. Although this reasoning chain leads to the correct final answer, it makes some overly definitive assumptions without reasonable justification (e.g., “must have attributes that match both”). In such cases, GPT-4 can accurately identify these vague statements through meticulous analysis. Moreover, the comprehensive analysis helps address the growing uncertainty arising from the ambiguity in understanding commonsense questions. For example, in the final step, GPT-4 offers extensive explanations for assigning low \mathcal{S} scores, considering the limited information available to comprehend the question.

In summary, GPT-4 demonstrates promising improvements in LLM evaluation by providing more rigorous and in-depth explanations. While there is significant variation in its generated explanations for some challenging cases, these explanations are typically beneficial as they elaborate on understanding and interpretation based on the given information. This suggests the potential for using LLM explanations to enhance self-evaluation in future research.

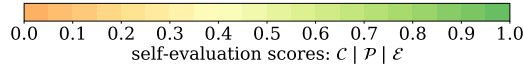


Figure 9: Score from low to high values visualized using colormap from orange to green.

⁵The results using GPT-4 were obtained on 22 April 2023.

[Q ₁] Grace weighs 125 pounds. Alex weighs 2 pounds less than 4 times what Grace weighs. What are their combined weights in pounds?	
[Ground-Truth a_1^*] 623.0	
[Predicted a_{12}] 627.0 	
[R_{12}] in Python	
<code>grace_weight = 125</code>	
\mathcal{E}	\mathcal{C} correct
\mathcal{S}	\mathcal{S}_1 correct
\mathcal{S}	\mathcal{S}_2 The line of code correctly assigns the given weight of Grace (125 pounds) to the variable <code>grace_weight</code>
\mathcal{S}	\mathcal{S}_3 The given line of code correctly assigns the value of 125 to the variable <code>grace_weight</code> , representing Grace's weight in pounds.
<code>alex_weight = 2</code>	
\mathcal{E}	\mathcal{C} <code>alex_weight</code> should be calculated based on <code>grace_weight</code>
\mathcal{S}	\mathcal{S}_1 <code>alex_weight</code> should be calculated as 4 times <code>grace_weight</code> minus 2
\mathcal{S}	\mathcal{S}_2 The line of code defining <code>alex_weight</code> should be calculated as 4 times Grace's weight minus 2 , which is <code>(4 * grace_weight) - 2</code> .
\mathcal{S}	\mathcal{S}_3 The calculation for Alex's weight is not correct. It should be calculated as <code>4 * grace_weight - 2</code> .
<code>weight_multiplier = 4</code>	
\mathcal{E}	\mathcal{C} correct
\mathcal{S}	\mathcal{S}_1 The line of code is correct as it defines the multiplier for Alex's weight based on Grace's weight.
\mathcal{S}	\mathcal{S}_2 Correct because it assigns the correct value (4) to the variable <code>weight_multiplier</code> , which represents the factor by which Grace's weight is multiplied to calculate Alex's weight.
\mathcal{S}	\mathcal{S}_3 The code correctly assigns the value 4 to the variable <code>weight_multiplier</code> , which will be used in the calculation of Alex's weight.
<code>alex_total = alex_weight + weight_multiplier * grace_weight</code>	
\mathcal{E}	\mathcal{C} correct formula to calculate <code>alex_total</code>
\mathcal{S}	\mathcal{S}_1 The formula used to calculate Alex's weight is incorrect.
\mathcal{S}	\mathcal{S}_2 The above line of code calculates Alex's weight correctly as 2 pounds less than 4 times Grace's weight <code>(4 * grace_weight - 2)</code> .
\mathcal{S}	\mathcal{S}_3 Alex weighs 2 pounds less than 4 times what Grace weighs, not 2 pounds more.
<code>answer = grace_weight + alex_total</code>	
\mathcal{E}	\mathcal{C} should be the sum of <code>grace_weight</code> and <code>alex_weight</code>
\mathcal{S}	\mathcal{S}_1 The code calculates Alex's weight correctly using the given relationship and adds it to Grace's weight to calculate the combined weight. <u>The only minor issue is the unnecessary use of the separate variable <code>alex_weight</code> which could directly be replaced with the calculation <code>(4 * grace_weight - 2)</code>. However, this does not affect the correctness of the answer.</u>
\mathcal{S}	\mathcal{S}_2 The result variable holds the correct combined weight of Grace and Alex, and this line of code correctly calculates the value of <code>answer</code> .
\mathcal{S}	\mathcal{S}_3 The code above correctly calculates the total weight of Grace and Alex based on the given problem statement. <u>The only minor issue is that the <code>alex_weight</code> variable is initialized as 2, which might be a bit confusing. It would be better to directly calculate <code>alex_total</code> as <code>(4 * grace_weight - 2)</code>, but the current code does not produce an incorrect result.</u>

Figure 10: Comparison between Codex (\mathcal{C}) and GPT-4 (\mathcal{S}) on evaluating the reasoning chains in arithmetic reasoning. The example (R_{12} for Q_1) is the same one from Figure 5a. We underline the key statements in explanations that justify the judgment of GPT-4.

[Q ₄] Is Freya a combination of Athena and Aphrodite?		
[Ground-Truth a_4^*] yes		
[Explanation] Athena was the Greek goddess of war. Aphrodite was the Greek goddess of love. Freya was the Norse goddess of war, love, and fertility.		
[Predicted a_{42}] yes ✓		
[R ₄₂] in Chain-of-Thought		
Freya is the goddess of love, beauty and fertility.		
ε	C	correct
S	S ₁	Freya indeed is associated with love, beauty, and fertility in Norse mythology. However, it is important to note that while the comparison to Athena and Aphrodite is understandable, it is <u>not a direct or complete equivalence</u> .
	S ₂	The statement correctly identifies Freya. However, the statement <u>does not mention that</u> Freya is a Norse goddess, which could slightly impact the understanding of the comparison to Greek goddesses Athena and Aphrodite.
	S ₃	The statement is mostly correct as Freya is indeed a goddess associated with love, beauty, and fertility. However, it might be worth mentioning that she is a Norse goddess, while Athena and Aphrodite are Greek goddesses, <u>to provide more context</u> .
You can compare these to the Greek gods Athena and Aphrodite.		
ε	C	correct
S	S ₁	It is true that Freya can be compared to the Greek gods Athena and Aphrodite in terms of their respective domains. However, the comparison is not entirely accurate as Freya comes from Norse mythology, and the attributes of the gods may not be an exact match. <u>The comparison is a simplification</u> , which is why the probability is not higher.
	S ₂	The statement establishes a comparison between Freya and the Greek goddesses Athena and Aphrodite. While this step of reasoning is generally correct, it could be improved by specifically mentioning that Freya is a Norse goddess and by <u>providing more information about the roles and attributes of Athena and Aphrodite for a clearer comparison</u> .
	S ₃	While it is true that Freya shares some similarities with Athena and Aphrodite, the comparison is not entirely accurate. Freya is a Norse goddess, while Athena and Aphrodite are Greek goddesses. The comparison is correct in the sense that they share some domains, but it might be <u>misleading</u> to suggest that Freya is a <u>direct "combination"</u> of these two goddesses.
It is not clear how their powers exactly compare.		
ε	C	correct
S	S ₁	This step of reasoning acknowledges that it is not entirely clear how the powers of Freya, Athena, and Aphrodite compare. Although there are some similarities, the differences between Norse and Greek mythologies make it difficult to determine whether Freya is a direct combination of Athena and Aphrodite. <u>The high probability reflects the acknowledgment of the uncertainty in comparing the powers of the gods across different mythologies</u> .
	S ₂	The statement acknowledges that it is not clear how the powers of Freya, Athena, and Aphrodite exactly compare. However, the question asks if Freya is a combination of Athena and Aphrodite, which implies a synthesis or blending of their roles or attributes. The reasoning could be improved by <u>specifically mentioning the roles and attributes of Athena and Aphrodite, and whether or not they overlap with Freya's roles and attributes</u> .
	S ₃	The reasoning here is more accurate, acknowledging that while there are similarities between Freya and Athena and Aphrodite, it is not clear how their powers exactly compare. It is important to consider that Freya is a Norse goddess and Athena and Aphrodite are Greek goddesses, so <u>their roles, domains, and characteristics may not be directly comparable</u> .
However, it seems that Freya must have attributes that match both.		
ε	C	correct
S	S ₁	incorrect because <u>it assumes that Freya must have attributes</u> that match both Athena and Aphrodite. While there may be some similarities in their attributes, it is not appropriate to conclude that Freya is a combination of the two Greek goddesses. Norse and Greek mythologies are distinct, and the deities in these pantheons may have different origins and roles.
	S ₂	The statement acknowledges that Freya has attributes that match both Athena and Aphrodite. However, it would be more accurate to <u>discuss the specific attributes</u> of Athena and Aphrodite to determine the extent of the similarities.
	S ₃	It assumes that Freya must have attributes that match both Athena and Aphrodite, but this may not be the case. While there are similarities between these goddesses, they are from different mythologies (Freya – Norse, Athena & Aphrodite – Greek), and their roles, domains, and characteristics may not be directly comparable or combined. <u>It is not accurate to assume that Freya is a combination of Athena and Aphrodite without a more in-depth analysis in their respective mythologies</u> .
So the answer is yes.		
ε	C	correct
S	S ₁	This step of reasoning is incorrect because it is just based on some similarities in their attributes. While there may be some similarities, <u>it is not accurate to say that Freya is a combination of the two Greek goddesses</u> . Norse and Greek mythologies are distinct, and the deities in these pantheons have different origins and roles.
	S ₂	Although Freya shares some attributes with both Athena and Aphrodite, it is an <u>oversimplification</u> to claim that she is a combination of the two. The <u>question seems to imply</u> that Freya might be a <u>direct mix of the two</u> , but this is not accurate.
	S ₃	The conclusion that Freya is a combination of Athena and Aphrodite is <u>not accurate based on the information provided</u> . While there are similarities between these goddesses, they belong to different mythologies (Freya is Norse, Athena and Aphrodite are Greek), and their roles, domains, and characteristics may not be directly comparable or combined.

Figure 11: Comparison between Codex (C) and GPT-4 (S) on evaluating the reasoning chains in commonsense reasoning. The example (R_{42} for Q_4) is the same one from Figure 5b. We underline the key points in GPT-4 rationales that explain the detailed understanding and analysis on the steps.

511 A.3 Implementation Details

512 Similar to beam search, we maintain k distinct candidates in the beam and sample n completions for each one.
 513 Thus, for each reasoning step s^t , the search space has a size of $k \cdot n$. After acquiring $k \cdot n$ samples, we retain
 514 k candidates by sampling from P_{beam} as Eq. 4. We set $k = 5, n = 16$ in practice to balance the quality and
 515 efficiency. The maximum number of steps to decode is capped at 16.

516 We set hyperparameters differently for various tasks and baselines. First, regarding the generation temperature
 517 γ , for arithmetic and symbolic reasoning with PAL using deterministic beam search ($\tau = 0.0$), we find that
 518 $\gamma \in [0.4, 0.8]$ generally works well. In contrast, for commonsense reasoning with CoT, a lower temperature
 519 ($\gamma \in [0.1, 0.5]$) is more effective, likely due to the increased randomness from the free-text format. In majority
 520 voting, higher γ is preferred to better explore the search space in reasoning, *e.g.*, $\gamma \geq 0.5$ for arithmetic reasoning.
 521 To further introduce sampling randomness in stochastic beam search for majority voting on multiple reasoning
 522 chains, we use $\alpha = 0.5$ for all datasets but different values of τ for each task. Specifically, we choose $\tau = 0.5$
 523 for PAL and $\tau = 0.2$ for CoT, as PAL typically decomposes the reasoning problem into more steps than CoT.
 524 Here we tune τ instead of α to be smaller in CoT as CoT naturally contains more randomness due to its free-text
 525 formulation as we observe in practice, where a smaller τ is more efficient to balance this randomness.

526 In previous works, majority voting on multiple reasoning chains involves sampling N (usually ≥ 20) reasoning
 527 chains and conducting a vote to determine the final answer, which can be time-consuming. In our approach,
 528 we simply perform majority voting among the N candidates in the last step of beam search from only a few
 529 times (≤ 10) of searching. As a result, our method does not introduce additional time complexity compared to
 530 the conventional majority voting method, although we sacrifice some diversity in the final outcomes due to the
 531 similarity of candidates within a beam.

532 **Prompts** We show examples of both the generation and evaluation prompts we use on different tasks in the
 533 following tables, where we only show one instance for each case. Full prompts and detailed formulations can be
 534 found in our code.

Table 5: Examples of few-shot exemplars of generation and evaluation CoT prompts on GSM8K. We adopt 8-shot and 5-shot learning for generation and evaluation, respectively.

GENERATION

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A:

There are 15 trees originally.

And there were 21 trees after some more were planted.

So $21 - 15 = 6$ trees were planted.

So the answer is 6.

EVALUATION

Q: Allison brought some CDs online. Each CD cost \$7. There was an additional charge of \$4 per order for shipping costs. The total bill came to \$60. How many CDs did Allison buy?

A:

Each CD cost 7 dollars.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is (A)

And there was an additional charge of 4 dollars.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is (A)

So the total cost of each CD is $7 + 4 = 11$ dollars.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is (B), because each CD costs 7 dollars.

So $60 / 11 = 5.45$.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is (B), because it cost 4 dollars for shipping costs. So the cost of CDs is $60 - 4 = 56$ dollars. So Allison bought $56 / 7 = 8$ CDs.

So the answer is 5.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is (A), but the value of the number of CDs is incorrect.

Table 6: Examples of few-shot exemplars of generation and evaluation PAL prompts on **GSM8K**. We adopt 9-shot and 5-shot learning for generation and evaluation, respectively.

GENERATION

Q: Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left?

```
def solution():
    money_initial = 23
    bagels = 5
    bagel_cost = 3
    money_spent = bagels * bagel_cost
    money_left = money_initial - money_spent
    result = money_left
    return result
```

EVALUATION

Q: A piece of square paper has a perimeter of 32 centimeters. Nicky’s dog, Rocky, tore off 1/4 of the paper. What is the area of the remaining paper?

```
def solution():
    perimeter = 32
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    fraction_torn = 1 / 4
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    area_total = (perimeter / 4) ** 2
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), because the total area of the square
    ↪ can be calculated by (perimeter / 4) ** 2
    area_remaining = (1 - fraction_torn) * area_total
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    result = area_total
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (B), because the result should be
    ↪ area_remaining
    return result
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), but the value of result is incorrect
```

Table 7: Examples of few-shot exemplars of generation and evaluation PAL prompts on AQuA. Here ... represents the same evaluation script as those in the previous steps. We adopt 8-shot and 5-shot learning for generation and evaluation, respectively.

GENERATION

Question: In a flight of 600 km, an aircraft was slowed down due to bad weather. Its average speed for the trip was reduced by 200 km/hr and the time of flight increased by 30 minutes. The duration of the flight is:

Answer Choices: A)1 hour; B)2 hours; C)3 hours; D)4 hours; E)5 hours

```
def solution():
    duration = Symbol('duration', positive=True)
    delay = 30 / 60
    total_disntace = 600
    original_speed = total_disntace / duration
    reduced_speed = total_disntace / (duration + delay)
    solution = solve_it(original_speed - reduced_speed - 200, duration)
    duration = solution[duration]
    result = duration
    return result
```

EVALUATION

Question: Two trains of length 150 m and 200 m are 100 m apart. They start moving towards each other on parallel tracks, at speeds 54 kmph and 72 kmph. In how much time will the trains cross each other?

Answer Choices: A)100/7 sec; B)80/7 sec; C)57/7 sec; D)110/7 sec; E)50/7 sec

```
def solution():
    train_1_speed = 54 / 60
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    train_2_speed = 72 / 60
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    distance_between_trains = 100
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    train_1_length = 150
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    train_2_length = 200
    # ...
    # The above line of code is: (A)
    time_to_cross = distance_between_trains / (train_1_speed +
    ↪ train_2_speed)
    # ...
    # The above line of code is: (B), because to cross each other, the
    ↪ total distance should also contain the train length
    result = time_to_cross
    # ...
    # The above line of code is: (B), because the final result should be in
    ↪ seconds, and the value of time_to_cross is incorrect
    return result
    # ...
    # The above line of code is: (A), but the value of result is incorrect
```

Table 8: Examples of few-shot exemplars of generation and evaluation PAL prompts on **SVAMP** and **ASDiv**. Here we utilize the same prompts as they have the same task formulation. We adopt 7-shot and 5-shot learning for generation and evaluation, respectively.

GENERATION

Passage: James bought 93 red and 10 blue stickers, he used 31 red sticker on his fridge and 7 blue stickers on his laptop.

Question: How many red stickers does James have?

```
def solution():
    original_red_stickers = 93
    used_red_stickers = 31
    red_stickers = original_red_stickers - used_red_stickers
    result = red_stickers
    return result
```

EVALUATION

Passage: A piece of square paper has a perimeter of 32 centimeters. Nicky’s dog, Rocky, tore off 1/4 of the paper.

Question: What is the area of the remaining paper?

```
def solution():
    perimeter = 32
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    side_length = perimeter / 4
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    area = side_length ** 2
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    result = area
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (B), because should calculate the
    ↪ remaining area after torn off as result
    return result
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), but the value of result is incorrect
```

Table 9: Examples of few-shot exemplars of generation and evaluation PAL prompts on **TabMWP**. We adopt 4-shot and 5-shot learning for generation and evaluation, respectively.

GENERATION

Table of "Coin collections":

Name | Number of coins
 Braden | 76 \ \ Camilla | 94 \ \ Rick | 86
 Mary | 84 \ \ Hector | 80 \ \ Devin | 83
 Emily | 82 \ \ Avery | 87

Question: Some friends discussed the sizes of their coin collections. What is the mean of the numbers?

```
def solution():
    number_of_coins_for_different_person = [76, 94, 86, 84, 80, 83, 82, 87]
    mean_of_the_numbers = sum(number_of_coins_for_different_person) /
    ↪ len(number_of_coins_for_different_person)
    result = mean_of_the_numbers
    return result
```

EVALUATION

Table of "Roller coasters per amusement park":

Stem | Leaf
 1 | 0, 0, 1, 6, 8, 9 \ \ 2 | 4, 4, 5, 7, 8, 8
 3 | 1, 2, 4, 4, 9, 9 \ \ 4 | 2, 3, 5, 6, 8, 9, 9

Question: Rodrigo found a list of the number of roller coasters at each amusement park in the state. How many amusement parks have fewer than 40 roller coasters?

```
def solution():
    number_of_roller_coasters_per_amusement_park = [10, 14, 14, 15, 16, 18, 19, 20, 24, 25, 26, 28, 29, 29, 29,
    ↪ 30, 34, 35, 36, 39, 40, 40, 40, 41, 42, 43, 44, 44, 45, 45, 46, 46, 47, 48, 48, 49, 49, 49, 50, 50, 51,
    ↪ 51, 52, 52, 53, 53, 54, 54, 55, 55, 56, 56, 57, 57, 58, 58, 59, 59, 60, 60, 61, 61, 62, 62, 63, 63, 64,
    ↪ 64, 65, 65, 66, 66, 67, 67, 68, 68, 69, 69, 70, 70, 71, 71, 72, 72, 73, 73, 74, 74, 75, 75, 76, 76, 77,
    ↪ 77, 78, 78, 79, 79, 80, 80, 81, 81, 82, 82, 83, 83, 84, 84, 85, 85, 86, 86, 87, 87, 88, 88, 89, 89, 90,
    ↪ 90, 91, 91, 92, 92, 93, 93, 94, 94, 95, 95, 96, 96, 97, 97, 98, 98, 99, 99]
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (B), beacuse values in the rows of Stem and Leaf represent the decimal and
    ↪ individual digits, respectively

    number_of_amusement_parks_with_fewer_than_40_roller_coasters = 0
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), because this is to initialize the
    ↪ number_of_amusement_parks_with_fewer_than_40_roller_coasters
    for number_of_roller_coasters in
    ↪ number_of_roller_coasters_per_amusement_park:
        if number_of_roller_coasters < 40:
            number_of_amusement_parks_with_fewer_than_40_roller_coasters +=
            ↪ 1
            # Is the above line of code:
            # (A) Correct
            # (B) Incorrect
            # The above line of code is: (A), but the value of
            ↪ number_of_roller_coasters_per_amusement_park is incorrect
    result = number_of_amusement_parks_with_fewer_than_40_roller_coasters
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), but the value of
    ↪ number_of_amusement_parks_with_fewer_than_40_roller_coasters is
    ↪ incorrect
    return result
    # ...
    # The above line of code is: (A), but the value of result is incorrect
```

Table 10: Examples of few-shot exemplars of generation and evaluation PAL prompts on Date Understanding from **Big-Bench**. We adopt 6-shot and 3-shot learning for generation and evaluation, respectively.

GENERATION

Q: 2015 is coming in 36 hours. What is the date one week from today in MM/DD/YYYY?

```
def solution():
    # If 2015 is coming in 36 hours, then today is 36 hours before.
    today = datetime(2015, 1, 1) - relativedelta(hours=36)
    # One week from today,
    one_week_from_today = today + relativedelta(weeks=1)
    # The answer formatted with %m/%d/%Y is
    result = one_week_from_today.strftime('%m/%d/%Y')
    return result
```

EVALUATION

Q: Yesterday was April 30, 2021. What is the date yesterday in MM/DD/YYYY?

```
def solution():
    # Yesterday was April 30, 2021.
    yesterday = datetime(2021, 4, 30)
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A)
    # Yesterday,
    yesterday = yesterday - relativedelta(days=1)
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (B), because the date of yesterday was
    ↪ already got and this is step is redundant
    # The answer formatted with %m/%d/%Y is
    result = yesterday.strftime('%m/%d/%Y')
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), but the value of yesterday is
    ↪ incorrect
    return result
    # Is the above line of code:
    # (A) Correct
    # (B) Incorrect
    # The above line of code is: (A), but the value of result if incorrect
```

Table 11: Examples of few-shot exemplars of generation and evaluation PAL prompts on Object Counting from **Big-Bench**. We adopt 4-shot and 1-shot for generation and evaluation, respectively.

GENERATION

Q: I have a drum, a flute, a clarinet, a violin, four accordions, a piano, a trombone, and a trumpet. How many musical instruments do I have?

```
def solution():
    musical_instruments_to_count = {
        'drum': 1,
        'flute': 1,
        'clarinet': 1,
        'violin': 1,
        'accordion': 4,
        'piano': 1,
        'trombone': 1,
        'trumpet': 1
    }
    num_musical_instruments = sum(musical_instruments_to_count.values())
    result = num_instruments
    return result
```

EVALUATION

Q: I have a peach, an onion, two garlicks, a nectarine, a yam, a carrot, a potato, a head of broccoli, a cabbage, a banana, and a stalk of celery. How many vegetables do I have?

```
def solution():
    # note: I'm not counting the peach, nectarine, banana, or celery as
    # ↪ vegetables
    vegetables_to_count = {
        'onion': 1,
        # ...
        # The above line of code is: (A)
        'garlic': 2,
        # ...
        # The above line of code is: (A)
        'yam': 1,
        # ...
        # The above line of code is: (A)
        'carrot': 1,
        # ...
        # The above line of code is: (A)
        'potato': 1,
        # ...
        # The above line of code is: (A)
        'broccoli': 1,
        # ...
        # The above line of code is: (A)
        'cabbage': 1
        # ...
        # The above line of code is: (A)
    }
    # ...
    # The above line of code is: (B), because celery is vegetable
    num_vegetables = sum(vegetables_to_count.values())
    # ...
    # ... (A), but the value of vegetables_to_count is incorrect
    result = num_vegetables
    # ...
    # ... (A), but the value of num_vegetables is incorrect
    return result
    # ...
    # ... (A), but the value of result is incorrect
```

Table 12: Examples of few-shot exemplars of generation and evaluation CoT prompts on **CSQA**. We adopt 7-shot and 3-shot learning for generation and evaluation, respectively.

GENERATION

Q: What do people use to absorb extra ink from a fountain pen?

Answer Choices: (a) shirt pocket; (b) calligrapher’s hand; (c) inkwell; (d) desk drawer; (e) blotter

A:

The answer must be an item that can absorb ink.

Of the above choices, only blotters are used to absorb ink.

So the answer is (e).

EVALUATION

Q: John’s took his appointment book with him to a checkup but couldn’t find it anywhere. He looked and he looked but he didn’t know where it was. It wasn’t until he bought a new one, went home, took a shower that he found it. Where was it?

Answer Choices: (a) office supply store; (b) wallet; (c) doctor’s office; (d) handbag; (e) pocket

A:

The answer must be a place where John could have put his appointment book.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (B), because it fails to reason out that John will check the place when taking a shower (taking off clothes)

Of the above choices, only wallet makes sense.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (B), because wallet is too small to put his appointment book, and it cannot be the only one which makes sense (pocket is also plausible)

So the answer is (b).

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (A), but the previous step is incorrect

Table 13: Examples of few-shot exemplars of generation and evaluation CoT prompts on **StrategyQA**. We adopt 6-shot and 4-shot learning for generation and evaluation, respectively.

GENERATION

Q: Do hamsters provide food for any animals?

A:

Hamsters are prey animals.

Prey animals are food for predators.

Thus, hamsters provide food for some animals.

So the answer is yes.

EVALUATION

Q: While viewing "Scary Movie" is the viewer likely to experience an increase in adrenaline?

A:

Adrenaline is a hormone that is released when the body is under stress.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (A)

Scary movies are designed to make the viewer feel stressed.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (B), because Scary Movie is a film that is a comedy take on horror, intended to make viewers laugh

Thus, the viewer is likely to experience an increase in adrenaline.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (A), but the previous step is incorrect

So the answer is yes.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (A), but the previous step is incorrect

Table 14: Examples of few-shot exemplars of generation and evaluation CoT prompts on Sports Understanding on **Big-Bench**. We adopt 8-shot and 2-shot learning for generation and evaluation, respectively.

GENERATION

Q: Is the following sentence plausible?

“KYLE PALMIERI WAS CALLED FOR SLASHING.”

A:

Kyle Palmieri is a hockey player.

Being called for slashing is part of hockey.

So the answer is yes.

EVALUATION

Q: Is the following sentence plausible?

“ROBIN LEHNER LAUNCHED THE DESPERATION HEAVE”

A:

Robin Lehner is a hockey player.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (B), because Robin Lehner is a goaltender which is different from a player

Launching the desperation heave is part of hockey.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (B), because launching the desperation heave is for player, not goaltender

So the answer is yes.

Is the above step of reasoning:

(A) Correct

(B) Incorrect

The above step of reasoning is: (A), but the previous step is incorrect
