Rectifying Belief Space via Unlearning to Harness LLMs' Reasoning

Ayana Niwa[1] **Masahiro Kaneko**[1] **Kentaro Inui**[1,2,3]
[1]MBZUAI [2]Tohoku University [3]RIKEN
{Ayana.Niwa, Masahiro.Kaneko, Kentaro.Inui}@mbzuai.ac.ae

Abstract

Large language models (LLMs) can exhibit advanced reasoning yet still generate incorrect answers. We hypothesize that such errors frequently stem from ***spurious beliefs***, which are propositions the model internally considers true but are incorrect. To address this, we propose a method to rectify the belief space by suppressing these spurious beliefs while simultaneously enhancing true ones, thus enabling more reliable inferences. Our approach first identifies the beliefs that lead to incorrect or correct answers by prompting the model to generate textual explanations, using our *Forward-Backward Beam Search* (FBBS). We then apply unlearning to suppress the identified spurious beliefs and enhance the true ones, effectively rectifying the model's belief space. Empirical results on multiple QA datasets and LLMs show that our method corrects previously misanswered questions without harming overall model performance. Furthermore, our approach yields improved generalization on unseen data, suggesting that *rectifying a model's belief space* is a promising direction for mitigating errors and enhancing overall reliability.

1 Introduction

Large Language Models (LLMs) trained on massive corpora have demonstrated remarkable reasoning capabilities, even on complex tasks (Brown et al., 2020; Hartmann et al., 2023; Ruis et al., 2024). However, they still generate logically flawed or factually incorrect answers. One fundamental question is: why do they generate erroneous outputs, and how can we mitigate such errors?

We hypothesize that many of these mistakes arise from **spurious beliefs** embedded in the model. "Belief" is defined by any proposition the model internally considers true, whereas "knowledge" is required to be factually correct (Kassner et al., 2021; Richardson et al., 2022; Kassner et al., 2023). Crucially, LLMs do not merely acquire factual knowledge from training data; rather, they integrate and generalize multiple pieces of knowledge to form new beliefs, resulting in the **belief space**.

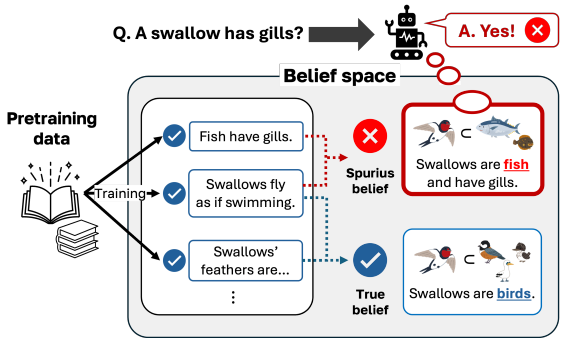


Figure 1: Example of a QA task: The model combines multiple beliefs from its training data to form new ones. If it references the spurious belief "*Swallows are fish and ...* ✗" during its reasoning, it may generate an incorrect answer. This study aims to suppress such spurious beliefs (✗), thereby allowing the model to draw on the true belief "*Swallows are birds* ✓" and ultimately avoid erroneous reasoning.

It is important to emphasize that beliefs are formed regardless of whether they are factually correct. Consider a conceptual example in Figure 1: a model might mistakenly combine the true belief "fish have gills ✓" with "swallows fly as if swimming ✓" and, could yield a spurious belief such as "*Swallows are fish and have gills* ✗." If the model references this spurious belief, it may incorrectly answer "*Yes*" to the question "*Do swallows have gills?*" (Kassner et al., 2021). This example illustrates how an LLM could form *incorrect implicit beliefs* that are not stated directly in the training corpus. Our ultimate goal is to rectify the belief space into a more **trusted space** by suppressing spurious beliefs and enhancing true ones (e.g., *Swallows are birds*" ✓), preventing incorrect inferences.

In this paper, we propose a framework to rectify the belief space by identifying the beliefs used for the reasoning, and then suppressing references

Findings of the Association for Computational Linguistics: ACL 2025, pages 25060–25075

to the spurious beliefs while enhancing references to the true beliefs. To identify the beliefs referenced by the model, we instruct it to explain the information required to generate the answer y from the given input text x. Specifically, we introduce a Forward-Backward Beam Search (FBBS) that maximizes both the forward likelihood (i.e., the plausibility of the belief given x) and backward likelihood (i.e., the probability of generating y from x and these beliefs) (Section 3.1). Subsequently, we apply unlearning based on gradient ascent (Yao et al., 2024; Liu et al., 2024b) to the identified beliefs to suppress references to spurious beliefs while giving priority to the true ones (Section 3.3). Through these steps, the model's belief space is more accurately reorganized, thereby reducing erroneous reasoning.

We demonstrate the effectiveness of our framework on multiple QA tasks (HotpotQA (Yang et al., 2018), SciQ (Welbl et al., 2017), and OpenBookQA (Mihaylov et al., 2018)) using three publicly available instruction-tuned LLMs (OLMo (Groeneveld et al., 2024), Pythia (Biderman et al., 2023), and RedPajama (Weber et al., 2024)). Compared to both the vanilla model (before our method) and baseline approaches that either suppress the incorrect answers themselves or knowledge in the training data, our method improves accuracy by up to 6.4 points for OLMo, 5.2 points for Pythia, and 8.0 points for RedPajama. Moreover, on unseen evaluation data, it achieves gains of up to 9.6 points for OLMo, 7.1 points for Pythia, and 8.4 points for RedPajama, underscoring its strong generalization capability. These results notably surpass the vanilla model's performance, indicating that *rectifying the belief space* can substantially enhance the model's reasoning. Furthermore, suppressing or enhancing beliefs does more than simply target individual beliefs; it effectively reorganizes the entire belief space to reduce errors and improve overall generalizability.

2 Beliefs in LLMs

2.1 Definition of Beliefs

Following prior work (Kassner et al., 2021; Richardson et al., 2022; Kassner et al., 2023), we define a **belief** in an LLM as a proposition that the model *considers* to be true, regardless of whether it is factually correct. Unlike knowledge, which is generally treated as necessarily factual, beliefs can be erroneous. We refer to the model's entire collection of such propositions as its *belief space*, denoted by $\mathcal{B}$.

Let $\mathcal{S}$ be the set of all propositions expressed in natural language. We introduce a function $\Gamma : \mathcal{S} \to \{\text{True}, \text{False}\}$ to determine whether an LLM considers any proposition $b \in \mathcal{S}$ to be true.

$$\mathcal{B} = \{\, b \in \mathcal{S} \mid \Gamma(b) = \text{True} \}, \tag{1}$$

$$\Gamma(b) = \begin{cases} \text{True} & \text{(if the LLM considers } b \text{ true)}, \\ \text{False} & \text{(otherwise)}. \end{cases} \tag{2}$$

Any belief $b \in \mathcal{B}$ defined in this way can be categorized into the following two types:

(1) Explicit Beliefs These are propositions that appear directly in the training data and are internalized by the model as-is. Indeed, numerous studies have shown that LLMs can memorize parts of their training data (Wang et al., 2025; Chen et al., 2024), and such memorized content is retained as explicit beliefs within the model.

(2) Implicit Beliefs These are propositions that the model internally reconstructs by combining pieces of information or performing analogical reasoning. For instance, as shown in Figure 1, the model might derive the belief "*Swallows are fish and ...* ✗" by combining information such as "*Fish have gills* ✓" and "*Swallows fly as if swimming* ✓." Previous work has demonstrated that LLMs are capable of integrating multiple pieces of knowledge for the inference (Treutlein et al., 2024). Crucially, even if the original training data is correct, the model may arrive at an spurious belief, leading to incorrect answers.

2.2 Reasoning Based on Beliefs

When given an input x (e.g., a question) and generating an output y (e.g., an answer), an LLM references some subset $\mathcal{B}_x \subseteq \mathcal{B}$ of the overall belief space that it deems necessary to answer x. From this subset, the model then chooses the most appropriate text y (i.e., performs inference). Formally:

$$y^* = \arg\max_y P\big(y \mid x, \mathcal{B}_x\big). \tag{3}$$

We denote by $\mathcal{B}_{x \to y} \subseteq \mathcal{B}_x$ the set of beliefs that actually contribute to generating the output y given the input x. In many cases, if $\mathcal{B}_{x \to y}$ is factually correct, the model arrives at a correct answer; if

$\mathcal{B}_{x\to y}$ is spurious, it yields an incorrect answer. Hence, to reduce erroneous reasoning, we seek to suppress **spurious beliefs** that lead to mistakes, and rectify the model's belief space into a more accurate trusted space.

3 Rectifying the Belief Space

We propose a two-phase procedure to rectify the belief space $\mathcal{B}$. First, we *identify* which beliefs the model relies on when it generates answers (Section 3.1 and 3.2). Second, we apply an *unlearning* step to suppress references to spurious beliefs while enhancing references to true ones (Section 3.3).

Here, we denote the spurious belief set as $\mathcal{B}^{\text{Spu}}_{x\to y_{\text{Inc}}}$ which leads to the incorrect answer y_{Inc}, and the true belief set as $\mathcal{B}^{\text{True}}_{x\to y_{\text{Cor}}}$ which yields the correct answer y_{Cor}.

3.1 Identifying Beliefs

Consider a given input-output pair (x, y) and the task of identifying the beliefs $\mathcal{B}_{x\to y}$ used to derive y from x. Previous research has typically provided candidate beliefs to the model and checked whether the model deems these beliefs true (Kassner et al., 2023). However, such an approach does not directly capture $\mathcal{B}_{x\to y}$, the set of beliefs specifically used in the inference process from x to y.

To address this, we adopt an approach based on *explanations*. That is, we prompt the model itself, under parameters $\boldsymbol{\theta}$, to explain which beliefs are necessary to derive y from x, thereby obtaining the belief set $\mathcal{B}^{*}_{x\to y}$. Specifically, we adopt a prompt that includes the input x and the output y, but leaves a blank (represented as _____). By generating the text that fills in this blank, we can acquire the beliefs $\mathcal{B}^{*}_{x\to y}$ that the model itself deems necessary to derive x to y. The prompt is as follows[1]:

```
{INPUT} The concise fact to solve the
problem is that _____. Therefore, the
answer is {OUTPUT}.
```

We replace {INPUT} with x in the portion preceding the blank to form a prefix prompt x^{pre}, and replace {OUTPUT} with y in the portion following the blank to form a suffix prompt y^{suf}. To obtain the spurious belief set $\mathcal{B}^{\text{Spu}}_{x\to y_{\text{Inc}}}$, we use the prompt by inserting the model's actual incorrect answer y_{Inc} into the {OUTPUT} slot. Similarly, for the true belief set $\mathcal{B}^{\text{True}}_{x\to y_{\text{Cor}}}$, we substitute the correct answer y_{Cor} into {OUTPUT} slot.

[1]As mentioned earlier, a belief does not necessarily correspond to an actual fact, but since it is information the model itself considers true, we use the term "fact" in the prompt.

3.2 Forward-Backward Beam Search (FBBS) for Belief Generation

To generate the beliefs $b \in \mathcal{B}^{*}_{x\to y}$ that fills the blank in the prompt (Section 3.1), we must consider both a forward constraint (i.e., plausibility of continuing from x^{pre}) and a backward constraint (i.e., how likely y^{suf} would be generated from x^{pre} and a given belief b). Formally, we consider:

$$\begin{aligned}&\arg\max_{b} P\big(y^{\text{suf}}, b \mid x^{\text{pre}}; \boldsymbol{\theta}\big) \qquad (4)\\ &= \arg\max_{b} \underbrace{P\big(b \mid x^{\text{pre}}; \boldsymbol{\theta}\big)}_{\text{forward}} \cdot \underbrace{P\big(y^{\text{suf}} \mid x^{\text{pre}}, b; \boldsymbol{\theta}\big)}_{\text{backward}},\end{aligned}$$

where the first term assesses the plausibility of generating b from x^{pre} (forward), and the second term assesses how well y^{suf} is generated from given x^{pre} and b (backward). We achieve this via our proposed Forward-Backward Beam Search (FBBS), an extended version of beam search (see Figure 2).

We consider the case of generating a belief $b \in \mathcal{B}^{*}_{x\to y}$ of length T, which is represented as $(b_1, b_2, \ldots, b_T)$. For simplicity, here we denote x^{pre} as x and y^{suf} as y in this section.

In standard beam search, the next token b_t is chosen by maximizing:

$$b_t \leftarrow \arg\max_{b_t} \log P\big(b_t \mid x, b_{<t}; \boldsymbol{\theta}\big), \qquad (5)$$

where $b_{<t}$ is the partially generated token sequence $(b_1, \ldots, b_{t-1})$. However, standard beam search does not explicitly account for whether the final output y will be generated. FBBS overcomes this limitation by looking ahead and evaluating the likelihood of ultimately generating y.

Concretely, we repeat the following steps (1)–(4) for $t = 1, \ldots, T$, thereby identifying sequences b that lead to y from x with high probability.

- **(1) Candidate Selection Based on Token Probability** As in standard beam search, we obtain the top n token candidates $\{b_t^{(i)}\}_{i=1}^{n}$ for step t by their local conditional probabilities $P^{(i)}_{\text{fwd}} = P\big(b_t \mid x, b_{<t}; \boldsymbol{\theta}\big)$. We refer to $\log P^{(i)}_{\text{fwd}}$ as the **forward** score.

- **(2) Estimating the Probability of Generating y^{suf} via Lookahead** For each candidate $b_t^{(i)}$, we concatenate it with $(x, b_{<t})$ and greedily generate tokens until reaching the end of a sequence.

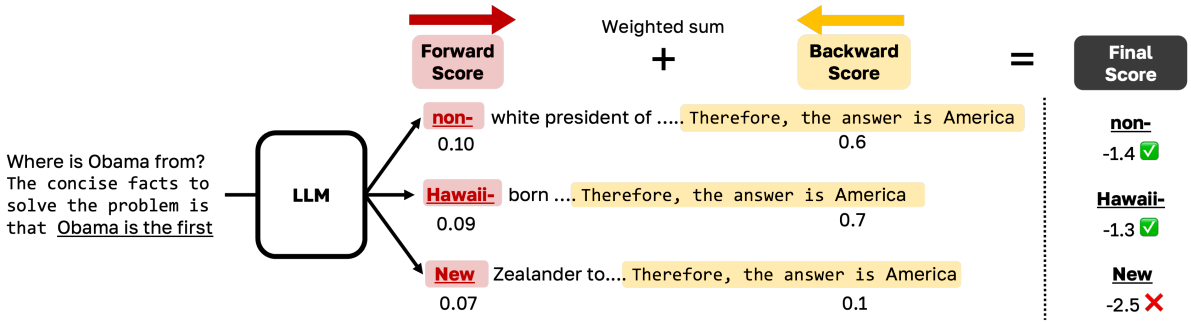


Figure 2: Forward-Backward Beam Search (FBBS) for generating beliefs. Shown here is the process of determining the next token among two candidates, "non-" and "Hawaii-", (i.e., $m = 2$) after partially generating the text "Obama is the first."

During this process, we measure the probability $P^{(i)}_{\text{back}} = P\big(y \mid x, b_{<t}, b^{(i)}_t; \boldsymbol{\theta}\big)$. We refer to $\log P^{(i)}_{\text{back}}$ as the **backward** score.

- **(3) Re-ranking Using a Weighted Score** We combine the forward score $\log P^{(i)}_{\text{fwd}}$ and the backward score $\log P^{(i)}_{\text{back}}$ as follows:

$$S^{(i)}_t = \lambda\big(t, \hat{T}^{(i)}_t\big) \cdot \underbrace{\log P^{(i)}_{\text{fwd}}}_{\text{forward}} + \big(1 - \lambda\big(t, \hat{T}^{(i)}_t\big)\big) \cdot \underbrace{\log P^{(i)}_{\text{back}}}_{\text{backward}}, \tag{6}$$

$$\lambda\big(t, \hat{T}^{(i)}_t\big) = \frac{1}{1 + \exp\Big(\alpha\Big(\frac{2t}{\hat{T}^{(i)}_t} - 1\Big)\Big)}, \tag{7}$$

where $\lambda\big(t, \hat{T}^{(i)}_t\big)$ is a function that dynamically shifts the weight from the forward score to the backward score as generation progresses and $\hat{T}^{(i)}_t$ is the sequence length generated in step (2). Specifically, early in the generation (small t), we emphasize the next token probability P_{fwd} to ensure coherent context; later in the generation (large t), we emphasize the lookahead probability P_{back} to ensure that the final output y is likely to be generated. The hyperparameter α controls the smoothness of the sigmoid function.

- **(4) Candidate Update** Based on the re-ranked scores $S^{(i)}_t$, we select the top $m\,(< n)$ tokens and proceed to generate b_{t+1}.

By applying the FBBS method to the input-output pairs (x, y_{Cor}) and (x, y_{Inc}), we can identify the respective beliefs $\mathcal{B}^{\text{True}*}_{x \to y_{\text{Cor}}}$ and $\mathcal{B}^{\text{Spu}*}_{x \to y_{\text{Inc}}}$.

3.3 Rectifying the Belief Space via Unlearning

Let $\boldsymbol{\theta}$ denote the parameters of a pretrained model with belief space $\mathcal{B}$. Suppose we aim to suppress the influence of the spurious beliefs $\mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}$ and to enhance the influence of the true beliefs $\mathcal{B}^{\text{True}}_{x \to y_{\text{Cor}}}$ within $\mathcal{B}$. When we denote by $\mathcal{B}_r = \mathcal{B} \setminus \mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}$, the remaining set of beliefs, the ideal parameters $\boldsymbol{\theta}^*_r$ that only retain $\mathcal{B}_r$ are obtained by:

$$\boldsymbol{\theta}^*_r = \arg\min_{\boldsymbol{\theta}} L(y, \mathcal{B}_r \mid x; \boldsymbol{\theta}), \tag{8}$$

where $L(\cdot)$ is a loss function. The goal of unlearning in this context is to obtain parameters $\boldsymbol{\theta}^*_r$ by effectively suppressing the spurious belief set $\mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}$, so that it makes easier to reference the true belief set $\mathcal{B}^{\text{True}}_{x \to y_{\text{Cor}}}$.

Concretely, we apply gradient *ascent* (Liu et al., 2024b) to the set of spurious beliefs $\mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}$. While standard gradient *descent* updates $\boldsymbol{\theta}$ to minimize $L(\boldsymbol{\theta})$, gradient *ascent* updates $\boldsymbol{\theta}$ in the reverse direction so as to maximize the loss. Generally, lowering the generation probability of a belief $\mathcal{B}_{x \to y}$ makes it more difficult for the model to reference that belief during inference of y, as indicated by Equation 3. Simultaneously, we explicitly enhance reference to the true beliefs $\mathcal{B}^{\text{True}}_{x \to y_{\text{Cor}}}$ (the beliefs that lead to a correct answer). Formally:

$$\boldsymbol{\theta}^*_r = \arg\max_{\boldsymbol{\theta}} \Big(\underbrace{\mathbb{E}_{b_i \in \mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}} [L(y_{\text{Inc}}, b_i \mid x; \boldsymbol{\theta})]}_{\text{suppress}} - \beta \underbrace{\mathbb{E}_{b_i \in \mathcal{B}^{\text{True}}_{x \to y_{\text{Cor}}}} [L(y_{\text{Cor}}, b_i \mid x; \boldsymbol{\theta})]}_{\text{enhance}} \Big), \tag{9}$$

where β balances suppressing $\mathcal{B}^{\text{Spu}}_{x \to y_{\text{Inc}}}$ and enhancing $\mathcal{B}^{\text{True}}_{x \to y_{\text{Cor}}}$. By performing this unlearning step, the model is guided toward a rectified belief space that avoids erroneous reasoning.

4 Experiments

In this study, we demonstrate that rectifying the belief space can reduce erroneous reasoning while preserving overall model performance.

4.1 Experimental Settings

First, the model $\boldsymbol{\theta}$ is executed on the task using the training data described later. Next, we rectify the belief space using our proposed method to obtain $\boldsymbol{\theta}_r$. Finally, we use $\boldsymbol{\theta}_r$ to perform inference and analyze the results. During inference, we employ standard beam search to generate the output text y for a given input x, without using beliefs.

Models We experiment with the following three instruction-tuned LLMs:

1. **OLMo (7B)**[2] (Groeneveld et al., 2024)
2. **Pythia (6.9B)**[3] (Biderman et al., 2023)
3. **RedPajama (7B)**[4] (Weber et al., 2024) (abbreviated as RPJ)

Datasets We focus on QA tasks that probe the model's belief, using HotpotQA (Yang et al., 2018) (free-form QA), SciQ (Welbl et al., 2017) (multiple-choice QA), and OpenBookQA (Mihaylov et al., 2018) (multiple-choice QA).[5] The LLM must have encountered each training instance from each dataset during its pretraining in order to unlearn them. Therefore, we first checked the pretraining corpus to verify that both the `question` and `answer` fields of each instance were completely included. Only those instances meeting this criterion were selected for the training set $\mathcal{D}_{\text{train}}$. The remaining instances are randomly split in equal proportions to create development $\mathcal{D}_{\text{dev}}$ and evaluation sets $\mathcal{D}_{\text{eval}}$. Hence, for each dataset, the sizes of $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{dev}}$, and $\mathcal{D}_{\text{eval}}$ are: HotpotQA: 70k, 4k, 4k; SciQ: 9k, 2k, 2k; OpenBookQA: 3k, 1k, 1k.

Baselines To mitigate erroneous reasoning, we explore several approaches. In addition to assessing the baseline performance (referred to as "Vanilla") before any modifications, we compare three distinct methods. Each method employs the unlearning process described in Equation 9, utilizing unique suppressing and enhancing sets:

- **Answer space rectifying (Answer-SR):** For a given question, we directly suppress the probability of generating an incorrect answer while enhancing that of the correct answer. This is the most straightforward approach to mitigate erroneous reasoning.
- **Knowledge space rectifying (Knowledge-SR):** For a given question, we suppress references to irrelevant knowledge and enhance the knowledge that supports the correct answer. The knowledge is identified from the training data. This method aims to prevent the model from incorrectly referencing the knowledge.
- **Belief space rectifying (Belief-SR) (Ours):** For a given question, we suppress references to spurious beliefs while enhancing the true beliefs that support the correct answer.

We now provide a more detailed account of Knowledge-SR. In our experiment, we define "knowledge" as the factually correct information directly contained in the training data, which we assume the model references as the basis for its reasoning. When the model's reasoning is incorrect, we assume that the training instances it relied on were referenced in error; these instances form our suppressing set in Equation 9. In contrast, our enhancing set represents the knowledge that should have been referenced. We extract it from the `evidence` field of each dataset, which is part of the model's pretraining corpus. To identify which pieces of knowledge the model uses, we apply a Training Data Attribution (TDA) method that finds the training instances most influential to the final outputs. Among several TDA techniques (Pruthi et al., 2020; Koh and Liang, 2017; Isonuma and Titov, 2024), we focus primarily on UnTrac-Inv (Isonuma and Titov, 2024) in this section, as it achieved the best performance in our preliminary experiments. Additional experimental details and results for other TDA methods are provided in the Appendix.

Evaluation Metrics We evaluate performance on both the training set $\mathcal{D}_{\text{train}}$ and the evaluation set $\mathcal{D}_{\text{eval}}$. As the evaluation metric, we use accuracy based on the exact match between the prediction and the reference across all datasets. Within $\mathcal{D}_{\text{train}}$, we distinguish:

[2] `allenai/OLMo-7B-Instruct`

[3] `allenai/open-instruct-pythia-6.9b-tulu`

[4] `togethercomputer/RedPajama-INCITE-7B-Instruct`

[5] In our experiments, we only use the `question` and `answer` fields of these datasets, and we do not utilize the `evidence` field, except for the TDA baseline method described in the "Baseline" paragraph.

Method	OLMo				Pythia				RPJ			
	$\mathcal{D}^{\text{✗}}_{\text{train}}$	$\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$	$\mathcal{D}^{\text{✗}}_{\text{train}}$	$\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$	$\mathcal{D}^{\text{✗}}_{\text{train}}$	$\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$
HotpotQA dataset												
Vanilla	0.0	100.0	93.1	42.9	0.0	100.0	86.9	34.3	0.0	100.0	87.1	36.5
Answer-SR	**92.6**	93.9	93.8	39.6	86.1	89.4	88.9	31.4	87.7	85.1	85.4	34.1
Knowledge-SR	81.0	89.6	89.0	42.9	83.7	85.6	85.3	33.5	86.9	84.0	84.3	35.6
Belief-SR (Ours)	86.6	**96.1**	**95.4**	**46.2**	**87.7**	**91.0**	**90.5**	**38.5**	**88.0**	**89.4**	**89.2**	**38.5**
SciQ dataset												
Vanilla	0.0	100.0	94.5	68.9	0.0	100.0	91.8	57.3	0.0	100.0	89.6	48.6
Answer-SR	90.6	91.1	91.0	62.0	88.5	92.0	91.7	55.0	89.0	91.0	90.7	44.2
Knowledge-SR	87.1	90.2	90.0	65.0	85.4	90.1	89.7	57.0	80.5	87.1	86.4	47.8
Belief-SR (Ours)	**92.8**	**95.4**	**95.2**	**71.4**	**91.7**	**93.4**	**93.2**	**60.2**	**89.3**	**91.4**	**91.1**	**52.6**
OpenBookQA dataset												
Vanilla	0.0	100.0	92.0	71.7	0.0	100.0	90.6	63.5	0.0	100.0	91.2	64.5
Answer-SR	88.3	90.5	90.3	65.8	83.1	88.2	87.7	59.3	85.8	86.4	86.3	61.7
Knowledge-SR	87.9	90.9	90.6	69.6	83.3	92.0	91.1	63.8	80.1	89.0	88.2	64.0
Belief-SR (Ours)	**93.5**	**94.7**	**94.6**	**75.4**	**90.4**	**92.0**	**91.8**	**66.0**	**93.3**	**94.5**	**94.3**	**68.2**

Table 1: Accuracy on three QA datasets. **Bold** indicates the highest score in each subset, and underlined marks scores statistically superior at $p = 0.01$ by bootstrap sampling compared to the second-best approach.

- $\mathcal{D}^{\text{✗}}_{\text{train}}$, the set of the training instances answered *incorrectly* by the vanilla model $\boldsymbol{\theta}$.
- $\mathcal{D}^{\text{✓}}_{\text{train}}$, the set of the training instances answered *correctly* by the vanilla model $\boldsymbol{\theta}$.
- $\mathcal{D}_{\text{train}}$, the entire training set.

Hyperparameters We adopt the following settings based on performance on the development set. In our Forward-Backward Beam Search (FBBS; Section 3.2), we use $\alpha = 0.3$ for the sigmoid-based dynamic weighting, a beam width of $n = 8$, and a candidate size of $m = 4$. For training, we use Adam with a learning rate of 5×10^{-5}, a batch size of 8, and $\beta = 0.5$ in Equation 9. When unlearning, we choose the belief b with the highest final score (Equation 6) as the target. We sample the same number of instances for the suppressing and enhancing set in Equation 9. During inference, we apply the default hyperparameters in the Transformers library (Wolf et al., 2020). For additional details, please refer to Appendix.

4.2 Results

Our main results are shown in Table 1.

Belief-Space Rectification Effectively Suppresses Erroneous Reasoning Let us begin with the results on the training data. We observe that across nearly all datasets, models, and baselines, our proposed method consistently improves accuracy on $\mathcal{D}^{\text{✗}}_{\text{train}}$, i.e., the previously misanswered instances. Compared with other rectification methods, it achieves improvements of up to 5.7 points for OLMo, 7.3 points for Pythia, and 13.2 points. Additionally, accuracy for the previously correct instances $\mathcal{D}^{\text{✓}}_{\text{train}}$ also increases relative to the baselines, leading to overall gains on the entire training set $\mathcal{D}_{\text{train}}$ (up to 6.4 points for OLMo, 5.2 for Pythia, and 8.0 for RPJ). Moreover, the updated model $\boldsymbol{\theta}_r$ by our proposed method outperforms the vanilla model $\boldsymbol{\theta}$ by up to 2.6 points for OLMo, 3.6 points for Pythia, and 3.1 points for RPJ. These results indicate that **rectifying the belief space can reduce incorrect reasoning without compromising the model's overall performance.**

Improving Belief Space Also Improves Generalization Turning to the results on the evaluation set $\mathcal{D}_{\text{eval}}$, the model $\boldsymbol{\theta}_r$ obtained by our proposed method outperforms both the baselines and the original vanilla model $\boldsymbol{\theta}$. Knowledge-SR methods that

suppress the knowledge (i. e., training instance) frequently degrade overall performance. This indicates that they can unintentionally eliminate valid information that is still useful for answering other questions. Our proposed approach avoids this pitfall by selectively suppressing only the "spurious beliefs" linked to incorrect reasoning, thus preserving necessary knowledge. When we compare our method to Answer-SR, we observe that Answer-SR can overfit to those specific instances and perform worse on $\mathcal{D}_{\text{eval}}$. In contrast, suppressing beliefs tied to those incorrect answers is more effective. This aligns with the established insight that jointly incorporating explanations can improve learning efficiency (Hartmann and Sonntag, 2022): by jointly unlearning the beliefs (explanations) associated with errors, we achieve better overall outcomes. In summary, belief-SR also excels in generalization. Rather than focusing on individual beliefs in isolation, this suggests that the model can **abstractly identify patterns of "what to forget" and reorganize the belief space**, thereby reducing errors without losing essential information. Additionally, we confirmed that performance on out-of-domain generalization was not degraded through cross-evaluations conducted by swapping evaluation datasets. For details, see Appendix A.2.5.

4.3 Analysis

Most Beliefs in the Model Are Newly Formed, Not Memorized To investigate differences between the pretraining corpus, we measure the n-gram overlap between the beliefs we identify and the model's pretraining corpus. Specifically, for each dataset, we measure the maximum n-gram matching to determine what percentage of the identified beliefs appears in the pretraining data. For the n-gram matching, we employed the high-speed engine, Infini-gram (Liu et al., 2024a). We also test whether the model's beliefs might simply be paraphrased from the training data. To do this, we take the training instances identified by the TDA method (UnTrac-Inv), paraphrase them using GPT-4 (OpenAI et al., 2024) and Claude 3 (Anthropic, 2024), then also measure their n-gram overlap with the pretraining corpus. Table 2 shows that the beliefs generated by our method overlap with the pretraining data at only 20%–30% for any dataset and model, whereas the paraphrased UnTrac-Inv samples exhibit overlaps of 60%–80% or more. This suggests that our method's beliefs are not merely memorized or paraphrased from the training data, but rather represent newly constructed information.

HotpotQA dataset			
	OLMo	Pythia	RPJ
Belief-SR (Ours)	30.4	20.1	27.2
Knowledge-SR	100.0	100.0	100.0
+ Para (GPT-4)	71.3	60.1	68.9
+ Para (Claude 3)	65.1	58.8	74.5
SciQ dataset			
	OLMo	Pythia	RPJ
Belief-SR (Ours)	27.7	23.6	30.1
Knowledge-SR	100.0	100.0	100.0
+ Para (GPT-4)	75.4	65.1	77.2
+ Para (Claude 3)	70.4	55.6	80.1
OpenBookQA dataset			
	OLMo	Pythia	RPJ
Belief-SR (Ours)	32.3	22.4	29.6
Knowledge-SR	100.0	100.0	100.0
+ Para (GPT-4)	80.1	59.9	79.6
+ Para (Claude 3)	73.8	59.1	77.4

Table 2: n-gram overlap ratios between model's pretraining data and spurious beliefs $\mathcal{B}_{\text{Spu}}$ (Ours) vs. knowledge in training data identified by Knowledge-SR, including GPT-4 and Claude 3 paraphrasings.

Forward-Backward Beam Search Delivers Better Overall Accuracy We generate beliefs using FBBS described in Section 3.2, which jointly optimizes both forward and backward constraints. To validate this approach, we compare it against three baseline methods for belief generation:

Post-Hoc Explanation Given input-output pairs (x, y), we prompt an LLM to generate the information needed to derive y from x.

Forward-only Beam Search (FBS) Uses only P_{fwd}, omitting the backward probability.

Backward-only Beam Search (BBS) Uses only P_{back}, omitting the forward probability.

We performed unlearning the beliefs generated by these baselines and compared their accuracies.

We display the results on HotpotQA in Figure 3 (for results on other datasets, see Appendix A.2.2). Our FBBS approach achieves the highest overall accuracy on both the training data ($\mathcal{D}_{\text{train}}$) and the evaluation data ($\mathcal{D}_{\text{eval}}$). While Post-Hoc Explanation sometimes achieves a higher accuracy on $\mathcal{D}^{\text{✗}}_{\text{train}}$ (instances previously misanswered), it tends to overfit those instances and degrades performance on the training and evaluation data overall. In con-

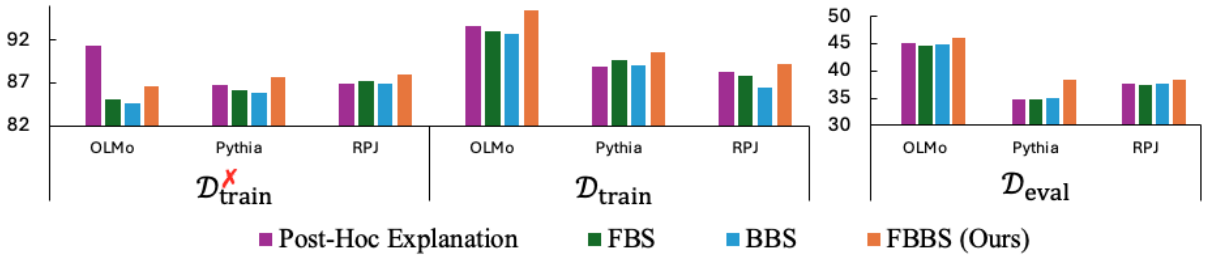


Figure 3: Comparison of accuracy on HotpotQA when unlearning beliefs generated by various generation methods.

Question:	*Which animal has the best camouflage in the Sahara?* *(A) a koala bear, (B) a horned viper, (C) Gyrfalcon, (D) a sloth*
Correct Answer: **Model Prediction:**	**(B) *A horned viper*** **(C) *Gyrfalcon***
Identified Knowledge **Identified Belief $\mathcal{B}_{\text{Spu}}$** (Ours)	*A desert environment contains very little food* *The gyrfalcon is commonly found in the **middle east** and is well-adapted to blending into the sahara's sandy terrain* ✗

Figure 4: An example from the OpenBookQA dataset with OLMo.

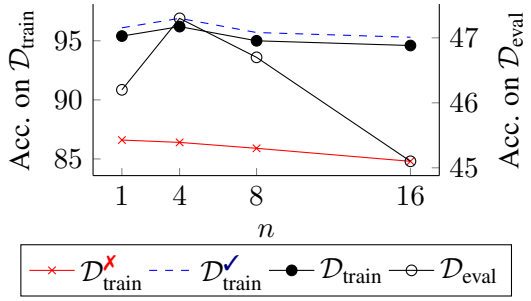


Figure 5: Change in accuracy when increasing the number of beliefs n used for unlearning (HotpotQA).

trast, FBBS provides a better balance by generating beliefs that are more broadly correctable.

The Top-1 Belief Sufficiently Represents the Belief Space We investigated whether increasing the number of identified beliefs (top-n, $n = 1, 4, 8, 16$) per instance used for rectification would yield additional performance gains. Figure 5 presents the results for the HotpotQA dataset (results for other datasets are provided in Appendix A.2.3). Increasing n does not lead to strictly monotonic improvements; rather, performance typically saturates around $n = 1, 4$. This is likely due to the additional noise introduced by considering too many beliefs. Our findings suggest that the top-1 belief already captures the most critical aggregated information from the belief space, and focusing on a small number of high-impact beliefs is sufficient for significant gains.

Qualitative Evaluation We evaluated the plausibility of beliefs generated by the OLMo model. For each dataset, we randomly sampled 50 spurious and 50 true beliefs, totaling 300 instances. Beliefs were assessed on four criteria, Consistency, Correctness, Conciseness, and Completeness, each rated on a 4-point scale (0–3), where higher scores indicate better evaluation results. Detailed descriptions of the metric are provided in Appendix A.2.4, and results are summarized in Table 3. True beliefs ($\mathcal{B}^{\text{True}}$) consistently achieved high scores (mean score of at least 2.3 out of a maximum of 3), indicating that the beliefs generated by our method represent plausible and accurate information relevant to answering the question. Conversely, spurious beliefs ($\mathcal{B}^{\text{Spu}}$) received substantially lower Correctness scores, less than half those of true beliefs across all datasets. These inaccuracies likely lead to incorrect answers due to errors within these beliefs.

Example of the Generated Beliefs Figure 4 presents an example from the OpenBookQA dataset using the OLMo model. The model incorrectly predicts that the "*gyrfalcon*," which actually inhabits Arctic regions, possesses the best camouflage in the Sahara. Knowledge-SR identifies a training instance mentioning merely that "*a desert environment contains very little food*," which fails to explain the model's wrong inference. In con-

	Consist.	Correct.	Concise.	Complete.
HotpotQA dataset				
$\mathcal{B}^{\text{Spu}}$	2.0	0.9	1.9	1.8
$\mathcal{B}^{\text{True}}$	2.3	2.0	2.2	2.3
SciQA dataset				
$\mathcal{B}^{\text{Spu}}$	2.2	0.7	2.3	2.2
$\mathcal{B}^{\text{True}}$	2.5	2.3	2.3	2.4
OpenBookQA dataset				
$\mathcal{B}^{\text{Spu}}$	2.3	1.1	2.1	2.1
$\mathcal{B}^{\text{True}}$	2.6	2.5	2.2	2.4

Table 3: Manual evaluation results.

trast, our method explicitly uncovers the spurious belief that "*The gyrfalcon is commonly found in the Middle East and ...*," thus revealing the internal misconception linking falcons to desert environments. As such, our approach more accurately pinpoints the faulty reasoning behind the model's error.

5 Related Work

5.1 Belief Editing in LLMs

The increased use of LLMs as knowledge bases has driven extensive research into editing the models' beliefs. Although existing studies typically use the term "knowledge editing," strictly speaking, these methods modify the LLM's beliefs rather than verified factual knowledge. Prominent approaches, referred to as knowledge editing (Wang et al., 2023b; De Cao et al., 2021; Meng et al., 2022), directly adjust model parameters to modify these internal beliefs without requiring full retraining. Alternative methods update the model's outputs using external editing networks (Mitchell et al., 2022) or constrained decoding to suppress outdated beliefs (Sun et al., 2024). However, these studies overlook the fact that the beliefs internally held by the model are not necessarily knowledge, that is, factually correct information. Our approach significantly differs by explicitly intervening in the model's "belief space," enabling more precise intervention into the model's actual reasoning process.

While several recent studies also address the beliefs of LLMs, their primary goal is often *belief coherence*, ensuring consistency among the beliefs, thus indirectly improving output consistency but not necessarily factual correctness (Kassner et al., 2023; Wang et al., 2023a; Jang et al., 2022; Kassner et al., 2021). In contrast, our research explicitly focuses on *belief factuality*, aiming to improve reasoning accuracy by directly rectifying spurious beliefs through unlearning. Another novel aspect is that, to achieve this, we enabled identification of beliefs directly linked to specific reasoning.

5.2 Process Supervision

Recent research has increasingly recognized the importance of explicitly supervising not only final outputs (*outcome supervision*) but also intermediate reasoning processes (*process supervision*). For instance, Lightman et al. (2024) showed that providing feedback for each intermediate reasoning step notably improves model performance compared to outcome-only supervision. Additionally, recent methods have leveraged fine-tuning approaches using explicit reasoning annotations, further enhancing model reasoning capabilities (Ho et al., 2023; Trung et al., 2024). Based on these findings, advanced models such as DeepSeek-R1 (DeepSeek-AI et al., 2025) explicitly include intermediate reasoning steps in their outputs, indirectly optimizing these steps through reinforcement learning. However, previous research on unlearning has largely neglected intermediate reasoning processes themselves. Our study addresses this gap by explicitly investigating the advantages of jointly unlearning beliefs (reasoning processes) alongside final answers. Our results confirm that this combined approach significantly improves the performance.

6 Conclusion

In this study, we proposed a method to rectify the belief space by **selectively suppressing references to spurious beliefs** that lead to erroneous reasoning and **enhancing references to true beliefs** in the belief space of an LLM. Specifically, we identify the beliefs used during inference by prompting the model to explain them, and then we apply unlearning. Our results demonstrate that our method effectively suppresses spurious beliefs that induce incorrect answers, raising the accuracy on previously misanswered instances without harming overall model performance. Moreover, we observed improved generalization on unseen data, highlighting the benefits of improving the correctness of the belief space itself. These findings show that rectifying the belief space offers a promising approach for both mitigating erroneous reasoning and enhancing the model's generalization performance.

7 Limitation

We introduced the Forward-Backward Beam Search (FBBS) method for generating the belief space of a pretrained model, demonstrating its effectiveness experimentally. However, because FBBS requires lookahead generation at each step, its computational cost is higher than conventional beam search. In practical applications, it would be desirable to develop more efficient search or approximation techniques to reduce this overhead.

Additionally, our experiments were conducted on datasets whose knowledge is contained in the model's training data, thus restricting the range of dataset-model combinations. Nonetheless, our approach to belief generation can, in principle, be applied to any model for which likelihood scores are available.

References

Anthropic. 2024. Introducing the next generation of claude.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tong Chen, Akari Asai, Niloofar Mireshghallah, Sewon Min, James Grimmelmann, Yejin Choi, Hannaneh Hajishirzi, Luke Zettlemoyer, and Pang Wei Koh. 2024. CopyBench: Measuring literal and non-literal reproduction of copyright-protected text in language model generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15134–15158, Miami, Florida, USA. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang,

Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Mareike Hartmann and Daniel Sonntag. 2022. A survey on improving NLP models with human explanations. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 40–47, Dublin, Ireland. Association for Computational Linguistics.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. Sok: Memorization in general-purpose large language models. *Preprint*, arXiv:2310.18362.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Masaru Isonuma and Ivan Titov. 2024. Unlearning traces the influential training data of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6312–6325, Bangkok, Thailand. Association for Computational Linguistics.

Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2022. BECEL: Benchmark for consistency evaluation of language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3680–3696, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Ashish Sabharwal, Kyle Richardson, Hinrich Schuetze, and Peter Clark. 2023. Language models with rationality. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14190–14201, Singapore. Association for Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024a. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Chris Liu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024b. Rethinking machine unlearning for large language models. *ArXiv*, abs/2402.08787.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch,

Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930. Curran Associates, Inc.

Kyle Richardson, Ronen Tamari, Oren Sultan, Dafna Shahaf, Reut Tsarfaty, and Ashish Sabharwal. 2022. Breakpoint transformers for modeling and tracking intermediate beliefs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9703–9719, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Laura Ruis, Maximilian Mozes, Juhan Bae, Siddhartha Rao Kamalakara, Dwarak Talupuru, Acyr Locatelli, Robert Kirk, Tim Rocktäschel, Edward Grefenstette, and Max Bartolo. 2024. Procedural knowledge in pretraining drives reasoning in large language models. *Preprint*, arXiv:2411.12580.

Zengkui Sun, Yijin Liu, Jiaan Wang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2024. Outdated issue aware decoding for factual knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9282–9293, Bangkok, Thailand. Association for Computational Linguistics.

Johannes Treutlein, Dami Choi, Jan Betley, Samuel Marks, Cem Anil, Roger B Grosse, and Owain Evans. 2024. Connecting the dots: Llms can infer and verbalize latent structure from disparate training data. In *Advances in Neural Information Processing Systems*, volume 37, pages 140667–140730. Curran Associates, Inc.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. ReFT: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614, Bangkok, Thailand. Association for Computational Linguistics.

Boshi Wang, Xiang Yue, and Huan Sun. 2023a. Can ChatGPT defend its belief in truth? evaluating LLM reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11865–11881, Singapore. Association for Computational Linguistics.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023b. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2025. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*.

Maurice Weber, Daniel Y Fu, Quentin Gregory Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Re, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

A Appendix

A.1 Further Experimental Details

Hyperparameter Selection Details We performed the hyperparameter search based on development set performance with the following candidate sets:

- **Forward-Backward Beam Search (FBBS):** The dynamic weighting parameter α was chosen from $\{0.3, 0.5, 0.7\}$, with the final value set to 0.3. The beam width and candidate size were fixed at $n = 8$ and $m = 4$, respectively.

- **Training:** The learning rate for Adam was selected from $\{1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}\}$ and set to 5×10^{-5}. The batch size was chosen from $\{1, 4, 8\}$ and set to 8, while the weight β in Equation 9 was chosen from $\{0.1, 0.5, 1\}$ and set to 0.5.

Details of TDA Ideally, TDA methods would examine the entire pretraining corpus to find the most influential training instances. However, this is computationally infeasible because the size of the pretraining corpus is massive. We therefore restrict the search space to the smaller, dataset-provided *evidence pool*, which still fully contains the relevant knowledge. These evidences are guaranteed to be part of the each model's pretraining data. We emphasize that this smaller evidence pool is of high quality, containing knowledge that is highly plausible as supporting evidence for the questions. Consequently, restricting TDA to this curated subset does not degrade the baseline TDA methods' performance.

A.2 Further Experimental Results

A.2.1 Overall Main Results

We present Table 4 showing all the results, including those obtained using the several TDA methods (**Grad-Dot (G-Dot)** (Pruthi et al., 2020), **Grad-Cos (G-Cos)** (Pruthi et al., 2020), **HIF** (Koh and Liang, 2017), **UnTrac (UT)** (Isonuma and Titov, 2024), and **UnTrac-Inv (UT-Inv)** (Isonuma and Titov, 2024). As a result, the effectiveness of our proposed method is still confirmed.

A.2.2 Comparison between Generation Methods

We present the entire results of comparing the multiple belief generation methods introduced in Section 4.3. The results for the HotpotQA dataset are shown in Figure 6, those for the SciQA dataset in Figure 7, and those for the OpenBookQA dataset in Figure 8.

	OLMo			Pythia			RPJ		
	$\mathcal{D}^{\text{✗}}_{\text{train}}/\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$	$\mathcal{D}^{\text{✗}}_{\text{train}}/\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$	$\mathcal{D}^{\text{✗}}_{\text{train}}/\mathcal{D}^{\text{✓}}_{\text{train}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$
HotpotQA dataset									
Vanilla	-	93.1	42.9	-	86.9	34.3	-	87.1	36.5
G-Dot	83.9/87.5	87.2	42.2	80.1/85.3	84.5	31.0	83.3/87.2	86.6	35.6
G-Cos	83.6/86.7	86.4	42.7	82.6/84.5	83.1	31.4	84.1/87.5	87.0	36.8
HIF	84.1/84.0	85.8	42.2	81.4/84.6	84.1	32.9	85.6/86.8	86.6	37.1
UT	82.2/88.4	87.9	42.8	83.3/86.1	85.7	34.1	87.4/83.7	84.1	36.3
UT-Inv	81.0/89.6	89.0	42.9	83.7/85.6	85.3	33.5	86.9/84.0	84.3	35.6
Ours	86.6/**96.1**	**95.4**	**46.2**	**87.7/91.0**	**90.5**	**38.5**	**88.0/89.4**	**89.2**	**38.5**
SciQ dataset									
Vanilla	-	94.5	68.9	-	91.8	57.3	-	89.6	48.6
G-Dot	84.9/86.7	86.6	63.0	87.9/91.9	91.5	57.4	85.0/85.4	85.3	45.9
G-Cos	88.3/90.1	90.0	65.9	82.7/89.2	88.6	56.1	83.2/85.1	84.9	45.0
HIF	89.0/91.0	90.8	66.0	82.0/90.4	89.7	56.4	79.7/85.8	85.1	45.2
UT	88.0/89.6	89.5	65.4	87.0/89.6	89.3	56.7	80.7/86.4	85.8	47.0
UT-Inv	87.1/90.2	90.0	65.0	85.4/90.1	89.7	57.0	80.5/87.1	86.4	47.8
Ours	**92.8/95.4**	**95.2**	**71.4**	**91.7/93.4**	**93.2**	**60.2**	**89.3/91.4**	**91.1**	**52.6**
OpenBookQA dataset									
Vanilla	-	92.0	71.7	-	90.6	63.5	-	91.2	64.5
G-Dot	85.0/88.9	88.5	66.7	80.9/89.5	88.6	62.6	79.5/88.6	87.7	62.4
G-Cos	86.3/90.1	89.7	67.0	85.3/88.3	88.0	62.6	80.2/87.3	86.6	61.3
HIF	87.5/90.7	90.4	67.8	84.0/90.2	89.6	62.9	79.6/87.5	86.8	61.4
UT	86.4/89.6	89.3	67.4	82.1/91.6	90.7	63.0	78.7/88.1	87.2	62.1
UT-Inv	87.9/90.9	90.6	69.6	83.3/92.0	91.1	63.8	80.1/89.0	88.2	64.0
Ours	**93.5/94.7**	**94.6**	**75.4**	**90.4/92.0**	**91.8**	**66.0**	**93.3/94.5**	**94.3**	**68.2**

Table 4: Accuracy on three QA datasets for all baselines.

A.2.3 Impact of the Number of Beliefs on Performance

Figure 9 shows how accuracy across all datasets changes when varying the number of beliefs n per example used for rectifying the belief space. In all datasets, performance did not monotonically increase with higher n, plateauing at $n = 1$ or $n = 4$.

A.2.4 Criteria of Qualitative valuation

The criteria used for the qualitative evaluation of beliefs identified by the proposed method are summarized in Table 5.

A.2.5 Cross-evaluation

To evaluate robustness of our proposed method against out-of-domain, we performed a cross-evaluation by swapping evaluation sets among HotpotQA, SciQ, and OpenBookQA. As shown in Table 6, we observed no significant degradation in performance even when the evaluation data was drawn from a different distribution (i.e., another domain). Naturally, to enhance effectiveness in a new domain, specialized methods such as domain adaptation or transfer learning (Zhuang et al., 2021) may be essential.

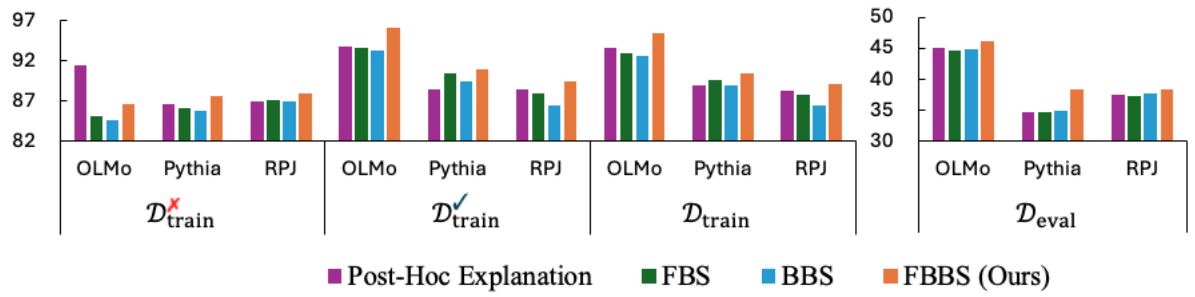


Figure 6: Comparison of belief generation methods on the HotpotQA dataset.

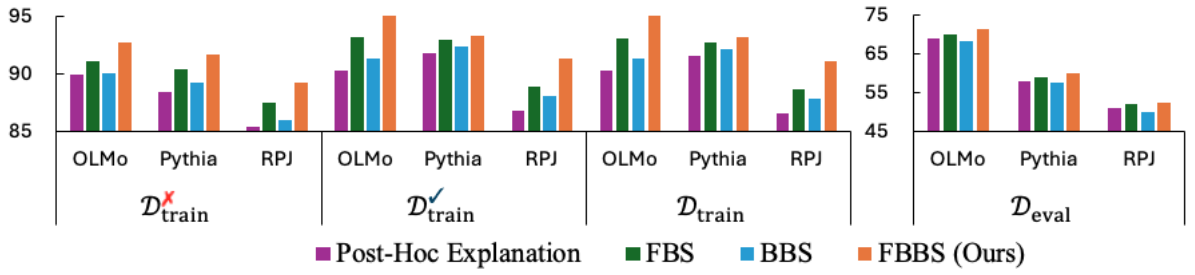


Figure 7: Comparison of belief generation methods on the SciQA dataset.

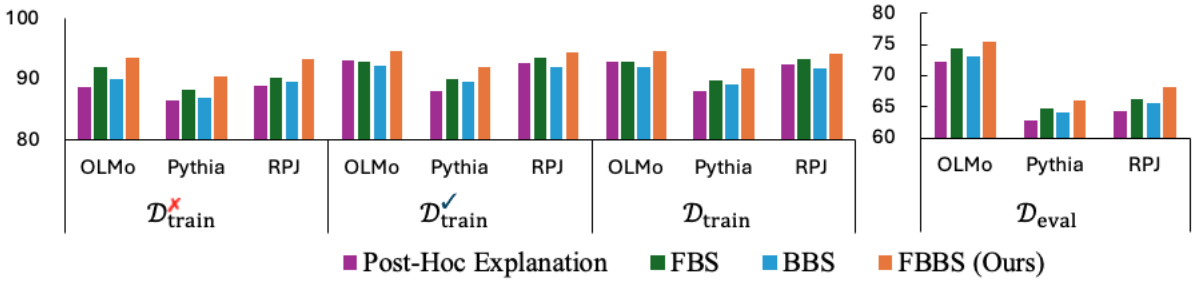
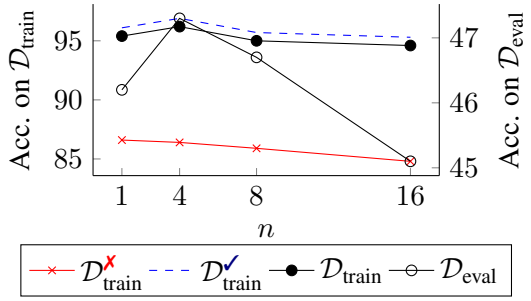


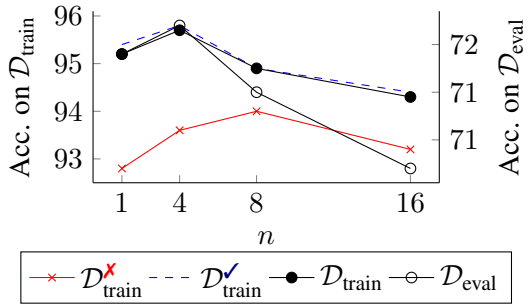
Figure 8: Comparison of belief generation methods on the OpenBookQA dataset.

Criteria	**3**	**2**	**1**	**0**
Consistency	Logically natural and consistent, with no leaps or contradictions.	Minor leaps or ambiguities, but the overall logic holds.	A clear logical leap or contradiction, making the reasoning insufficient.	Multiple logical leaps or contradictions, rendering the reasoning fundamentally flawed.
Correctness	All information is factually accurate.	Some information is ambiguous, but no clear factual errors.	Contains one clear factual error.	Contains multiple factual errors, making content generally unreliable.
Conciseness	Includes only necessary information; highly concise.	Slightly redundant, but does not hinder understanding.	Substantial redundancy or irrelevant content, impairing comprehension.	Severely redundant or off-topic, making reasoning difficult to understand.
Completeness	All necessary information included.	Some supplementary information missing, but conclusion still reachable.	Lacks important information required to support the conclusion.	Most necessary information missing, making conclusion unsupported.

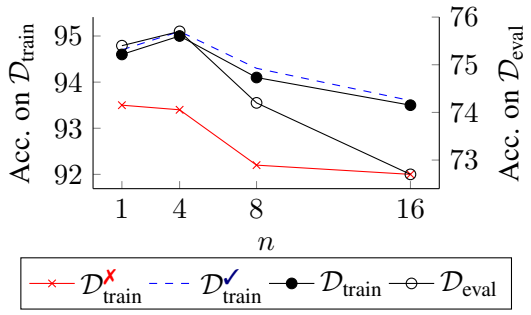
Table 5: Manual evaluation criteria for beliefs.



(a) HotpotQA



(b) SciQA



(c) OpenBookQA

Figure 9: Accuracy with different numbers of the beliefs.

Train \ Eval	**HotpotQA**	**SciQA**	**OpenBookQA**
HotpotQA	41.0	58.8	66.3
SciQA	38.6	61.4	67.2
OpenBookQA	38.4	59.1	69.8
Vanilla	37.9	58.2	66.5

Table 6: Evaluation results across different combination of training datasets and evaluation datasets.