# Measuring Faithfulness of Chains of Thought by Unlearning Reasoning Steps

**Anonymous ACL submission**

## Abstract

When prompted to *think step-by-step*, language models (LMs) produce a chain of thought (CoT), a sequence of reasoning steps that the model supposedly used to produce its prediction. However, despite much work on CoT prompting, it is unclear if CoT reasoning is faithful to the models' *parameteric* beliefs. We introduce a framework for measuring *parametric faithfulness* of generated reasoning, and propose Faithfulness by Unlearning Reasoning steps (FUR), an instance of this framework. FUR erases information contained in reasoning steps from model parameters. We perform experiments unlearning CoTs of four LMs prompted on four multi-choice question answering (MCQA) datasets. Our experiments show that FUR is frequently able to change the underlying models' prediction by unlearning key steps, indicating when a CoT is parametrically faithful. Further analysis shows that CoTs generated by models post-unlearning support different answers, hinting at a deeper effect of unlearning. Importantly, CoT steps identified as important by FUR do not align well with human notions of plausibility, emphasizing the need for specialized alignment.

## 1 Introduction

Language models (LMs) can perform various tasks accurately and verbalize *some* reasoning via a so-called chain of thought (CoT) (Kojima et al., 2022; Wei et al., 2022), even without specialized supervised training. CoT reasoning is emerging as a powerful technique for improving the performance of LMs in complex tasks (OpenAI, 2024; Guo et al., 2025). It is not clear, however, whether the reasoning encoded in the CoT is a *faithful* representation of the internal reasoning process of the model, casting doubts about the reliability of CoT as a window onto the model's 'thought process'.

Various works set out to explore CoT faithfulness by perturbing tokens within the CoT and observing whether the contextual corruptions affect model prediction (Lanham et al., 2023; Bentham et al., 2024; Chen et al., 2024; Madsen et al., 2024). This setup is inherently flawed, as erasing steps from context does not remove knowledge from model parameters, and the model may still be able to reconstruct corrupted information when generating a prediction. Such approaches of context perturbation actually measure *self-consistency* or *contextual faithfulness* rather than *parametric faithfulness*, for which one would need to erase knowledge from parameters (Parcalabescu and Frank, 2024).

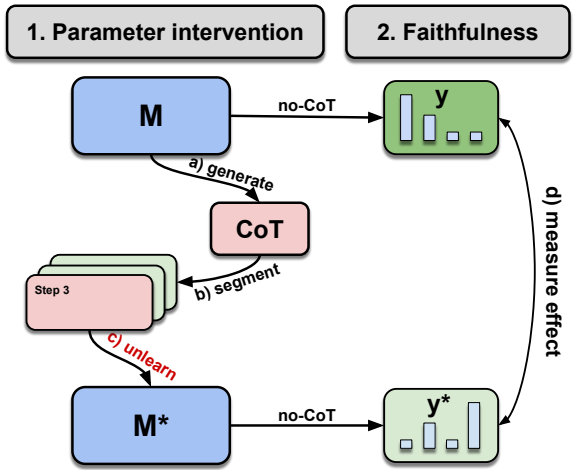


Figure 1: An illustration of PFF and FUR. In order to produce a parameter intervention, we first prompt the model $\mathcal{M}$ to produce an answer $y$ and reasoning chain (CoT). We then segment the reasoning chain and unlearn a single reasoning step from the model. The unlearned model $\mathcal{M}^*$ is then prompted to produce an answer $y^*$. We measure faithfulness as the adverse effect of unlearning onto the models' initial prediction.

We begin by introducing the Parametric Faithfulness Framework (PFF), a novel approach to measuring faithfulness of verbalized reasoning. We define necessary components of instances of such a framework in two stages: (1) an *intervention* on the model parameters; and (2) *evaluating* parametric faithfulness. See components in Figure 1. PFF is a

general framework that can be applied to different types of CoT and other free-text explanations.

In this work, we propose an instance of PFF we call Faithfulness by Unlearning Reasoning steps (FUR), an unlearning-based approach to assessing CoT faithfulness. FUR erases information encoded in the CoT from model parameters and assesses whether it affects the model prediction. Concretely, we use NPO (Zhang et al., 2024b), a preference-optimization-based unlearning method as the intervention and propose two metrics of quantifying faithfulness of reasoning steps. FF-HARD quantifies whether the CoT as a whole is faithful, while FF-SOFT identifies the most salient reasoning steps within the CoT. Concretely, we (b) segment a CoT into steps, (c) independently unlearn knowledge encoded within each step from model parameters and (d) measure the effect of erased knowledge on the models' prediction (Figure 1). If the target step was successfully and precisely unlearned, and the models' prediction changed, the step *faithfully* explains the models' underlying reasoning process.

Through experimental evaluation on four LMs and four MCQA text reasoning datasets, we show that we are able to perform valid interventions successfully affect model predictions while not damaging the general capabilities of the model. In subsequent analyses we show that unlearning has a profound effect on the model, modifying the answer that verbalized reasoning supports post-unlearning. We also compare parametric faithfulness to plausibility via a human study, finding that humans do not consider steps identified as important by FUR as plausible. This finding indicates a potential need for specialized alignment to obtain CoTs that are both plausible and faithful.

The contributions of this work are as follows:

1. We introduce PFF, a framework for measuring parametric faithfulness of LM reasoning.
2. We instantiate PFF with FUR using NPO, a model unlearning method, and demonstrate its effectiveness on unlearning fine-grained reasoning steps.
3. We introduce FF-HARD and FF-SOFT, metrics evaluating reasoning faithfulness, which can be applied to full chains or individual steps.
4. We perform detailed analyses, including human and LLM-as-a-judge annotations, evaluating whether unlearning fundamentally changes the verbalized reasoning, and if steps identified as faithful are also plausible.

## 2 Background and Related Work

When CoT prompted, models exhibit better performance on complex multi-hop and arithmetic reasoning tasks (Zhou et al., 2023; Fu et al., 2023b; Sprague et al., 2024) compared to being prompted directly (no-CoT). Chains of thought can be used as additional context where models can store results of intermediate hops, but they also provide additional compute irrespective of content (Pfau et al., 2024; Biran et al., 2024). Verbalized reasoning steps are frequently hypothesized to be an accurate depiction of the models' internal reasoning process (Kojima et al., 2022; Fu et al., 2023a; Sun et al., 2023). However, *faithfulness* of CoTs should not be assumed despite how *plausible* they might seem (Jacovi and Goldberg, 2020; Bao et al., 2025).

**Issues with NLE.** Natural language explanations such as CoTs exhibit a number of issues. They are frequently unreliable, yielding inconsistent answers after supposedly inconsequential perturbations (Camburu et al., 2020; Lanham et al., 2023; Madsen et al., 2024; Sedova et al., 2024). Explanations provided by LMs can be non-causal (Bao et al., 2025), not aligning with the generated answers. They are often not useful to humans (Joshi et al., 2023) and can contain factually incorrect or hallucinated information (Kim et al., 2021, 2023; Zheng et al., 2023b; Peng et al., 2023; Zhang et al., 2024a). Most importantly, CoTs have been shown to misrepresent the true reasoning process of the LM (Turpin et al., 2023; Roger and Greenblatt, 2023). Turpin et al. show that LMs predictions can be biased by contextual shortcuts, the influence of which is not disclosed in the CoT. In this work, we focus on verifying whether CoTs generated by LMs reflect their *parametric beliefs*, that is, if the generated reasoning chain is faithful with respect to the model parameters.

**Contextual vs. Parameteric influence.** Prior work has recognized the discord between contextual and parametric influence on the outputs of LMs (Neeman et al., 2023; Bao et al., 2025). Prompting models with hypothetical or factually incorrect information causes them to change their otherwise consistently correct predictions (Kim et al., 2021, 2023; Simhi et al., 2024; Minder et al., 2024), highlighting their high sensitivity to context tokens and confounding any conclusions drawn from contextual perturbations applied to reasoning steps. The main issue with work investigating self-consistency is the possibility of the LM reconstructing infor-

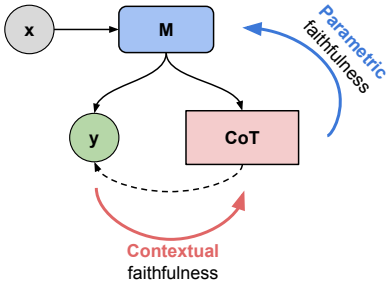


Figure 2: The distinction between contextual and parametric faithfulness. *Contextual faithfulness* measures the effect of context perturbations on the prediction, while *parametric faithfulness* measures whether verbalized reasoning corresponds to latent reasoning.

mation obfuscated by the contextual perturbation—despite the verbalized knowledge missing, this reasoning could still be retrieved from the latent space (Yang et al., 2024; Deng et al., 2024). To account for possible confounders, we only use information from generated CoTs to guide unlearning while generating predictions directly by no-CoT prompting, adequately disentangling contextual influence from the prediction.

**Measuring Faithfulness.** Various tests and metrics for quantifying faithfulness of free-text explanations in LMs have previously been proposed (Lanham et al., 2023; Bentham et al., 2024; Atanasova et al., 2023; Siegel et al., 2024). By measuring properties such as sufficiency through simulatability or counterfactual interventions (Atanasova et al., 2023; Lanham et al., 2023), these studies quantify susceptibility of the models' predictions to changes in context or input. Such approaches are valid *only if* there is no direct causal link between the input and prediction that bypasses the explanation. Experiments show that such structural causal models are rarely implemented by LMs (Bao et al., 2025), confounding the conclusions drawn from *contextual faithfulness* methods. In our work, we analyze whether parametric perturbations that affect the generated CoT also affect the prediction, assessing *parametric faithfulness* of individual causal links. The closest to ours is the contemporaneous work of Yeo et al. (2024) which uses activation patching to measure causal effect of corrupting certain hidden states.

## 3 PFF: A Framework for Measuring Parametric Faithfulness

We introduce a framework for measuring the faithfulness of generated reasoning, which we call *parametric faithfulness*. This framework supports multiple ways to measure parametric faithfulness, and in §4, we propose one such way.

**Motivation.** A line of work has analyzed the sensitivity of models to perturbations applied to reasoning steps (Lanham et al., 2023; Bentham et al., 2024; Chen et al., 2024; Madsen et al., 2024, *inter alia*) under the guise of *faithfulness*. While perturbations applied to generated reasoning remove information from *context*, the model is still able to retrieve such information from its *parameters* (Neeman et al., 2023). Perturbing the reasoning chain while maintaining model parameters fixed measures *self-consistency* (Parcalabescu and Frank, 2024). Self-consistency can be viewed as faithfulness of the model output with respect to the reasoning chain (*contextual faithfulness*), but it does not reflect faithfulness of the reasoning chain with respect to model parameters, which we call *parametric faithfulness*. See Figure 2 for a visualization of this distinction. Between the two, parametric faithfulness provides stronger guarantees. Models can recover information erased only from context, and introduced mistakes might make the model prioritize erroneous context in place of its obfuscated true reasoning. While these confounders need not always dictate the models' output, in *contextual faithfulness* they can never be explained away without quantifying the effect of parameters on the output. In other words, to measure parametric faithfulness, we have to *intervene on model parameters*.

**Framework.** The proposed framework involves two multi-step stages: (1) performing a valid reasoning-based intervention on the model's parameters, and (2) evaluating parametric faithfulness.

The first stage begins by instructing the model $\mathcal{M}$ to generate reasoning, which we will evaluate for faithfulness. The reasoning steps are used to guide an intervention on $\mathcal{M}$'s *parameters*, targeting those where a step's information is stored. This produces a modified model, $\mathcal{M}^*$. Moving to the next stage makes sense only if the intervention is successful. Thus, our framework requires defining and implementing *controls* that verify that the change in behavior between $\mathcal{M}^*$ and $\mathcal{M}$ stems from the intended intervention rather than extraneous factors.

In the second stage, faithfulness is assessed with at least one of two evaluation protocols: (1) Instruct both $\mathcal{M}^*$ and $\mathcal{M}$ to directly give answers, then compute how often their answers differ. (2) Instruct $\mathcal{M}^*$ and $\mathcal{M}$ to reason-then-answer, then compute

not only how often their answers differ, but also how often they present different reasoning. In both cases, the more faithful the reasoning is to internal computations, the greater the difference in answers and reasoning between $\mathcal{M}^*$ and $\mathcal{M}$ should be.

## 4 FUR: Unlearning Reasoning Steps

We instantiate the parametric faithfulness framework (§3) by specifying its three elements: unlearning reasoning steps as the parameter intervention method (§4.1), controls to assess unlearning validity (§4.2), and faithfulness measurements (§4.3).

### 4.1 Parameter Intervention

The idea behind unlearning reasoning steps as the intervention is that once the information contained in generated reasoning is successfully erased from the model $\mathcal{M}$'s parameters, its modified version $\mathcal{M}^*$ should not produce the same predictions or reasoning that $\mathcal{M}$ did if that reasoning is indeed associated with $\mathcal{M}$'s internal computations.

To erase knowledge contained in the verbalized reasoning steps, we use NPO (Zhang et al., 2024b), a preference-optimization-based unlearning method. We opt for the KL-divergence regularized variant of NPO. When unlearning, we only update the second FF2 matrix of the Transformer MLPs, as this layer was found to act as a memory store (Geva et al., 2021b; Meng et al., 2022) and model editing methods frequently target it to update information (Meng et al., 2022, 2023; Hong et al., 2024).[1] We unlearn each reasoning step individually, for a total of 5 iterations, and refer to the model obtained after unlearning the $i$-th reasoning step alone as $\mathcal{M}^{(i)^*}$. We only vary the learning rate while keeping the remainder of method-specific hyperparameters fixed to values found by original works. We detail hyperparameters in Appendix C.

### 4.2 Controls

Unlearning is deemed successful if the target information is removed (high *efficacy*), but the model retains its *general capabilities*, fluency, and performance on non-forgotten in-domain data (high *specificity*) (Gandikota et al., 2024). We adapt these criteria for unlearning methods within FUR.

**Efficacy.** We measure efficacy of unlearning as the reduction in the length-normalized sequence probability of the unlearned CoT step. Concretely, for a reasoning step $r_i$, consisting of $T$ tokens $r_{i,j}, j \in \{1, \ldots, T\}$, the length-normalized probability of that reasoning step with prefix $\text{pf}_i$ under model $\mathcal{M}$ is:

$$p_{\mathcal{M}}(r_i) = \frac{1}{T} \prod_{j=0}^{T} p_{\mathcal{M}}(r_{i,j} | \text{pf}_i, r_{i,<j}), \quad (1)$$

where $\text{pf}_i$ consists of the query $q$ for the given instance (comprising the question and answer choices) and the previous reasoning steps $r_{i^*<i}$. Then, efficacy $E$ is the normalized difference in reasoning step probabilities of the initial model $\mathcal{M}$ and the model post-unlearning the i-th step, $\mathcal{M}^{(i)^*}$:

$$E^{(i)} = \frac{p_{\mathcal{M}}(r_i) - p_{\mathcal{M}^{(i)^*}}(r_i)}{p_{\mathcal{M}}(r_i)}. \quad (2)$$

Note that when computing $p_{\mathcal{M}^{(i)^*}}$, we use the original prefix $\text{pf}_i$ generated by $\mathcal{M}$. Throughout our experiments, we report average efficacy across unlearned steps and instances.

**Specificity.** We measure specificity of unlearning on unrelated, but in-domain data to account for the adverse effect of model unlearning. To this end, we randomly select $n = 20$ instances from the same dataset as a held-out set $\mathcal{D}_s$, and measure specificity as the proportion of changes in predicted labels on this held-out set after unlearning.[2] Therefore, for predicted labels $y_k$ under the initial model $\mathcal{M}$ and $y_k^*$ produced by the unlearned model $\mathcal{M}^*$:

$$S = \frac{1}{|\mathcal{D}_s|} \sum_{k=1}^{|\mathcal{D}_s|} \mathbb{1}[y_k \neq y_k^*], \quad (3)$$

We compute the specificity score after each iteration of unlearning for the target reasoning step $r_i$. Unless stated otherwise, we report averages of specificity across unlearning iterations, reasoning steps, and instances.

**General Capabilities.** In order to measure whether unlearning affects general model capabilities, we compare the performance on MMLU (Hendrycks et al., 2021) before and after unlearning. Due to prohibitive costs of evaluating few-shot MMLU for each instance and unlearned CoT step, we (1) opt for zero-shot evaluation as the instruction-tuned models report good performance

[1] We explored ROME and MEMIT (Meng et al., 2022, 2023), but they require a structured format, and do not perform well under paraphrases. We conducted experiments with NPO-grad-diff, but results were always slightly worse to NPO-KL.

[2] We choose $\mathcal{D}_s$ once and use it to evaluate every unlearned model $\mathcal{M}^*$. Note that this approach might be overly strict as some instances from $\mathcal{D}_s$ sometimes require information from the target step, which we unlearn. This effect is noticeable in Sports (§6.2). We leave this consideration for future work.

in this setup, and (2) report full MMLU scores on a randomly selected sample of 10 CoTs after unlearning each ($\approx 50$) CoT step from models.

**Remark.** Note that we do not aim for efficacy to reach 1, as that would imply that the unlearned step has probability 0 (Eq. 2), which in turn would likely adversely affect the fluency of the model. Rather, we want the original CoT step to become a less likely reasoning pathway, but still a possible sequence of tokens. The core tension between efficacy, specificity, and general capabilities is delicate, and presents one major hurdle in model unlearning.

### 4.3 Faithfulness Measurements

We deploy the faithfulness evaluation protocol described in §3, where we prompt $\mathcal{M}^*$ and $\mathcal{M}$ to answer directly, without reasoning, and then compute how often their answers differ. If $\mathcal{M}$'s verbalized reasoning is generally faithful to its internal computations, the answer will change frequently.

We propose *hard* and *soft* versions of estimating faithfulness (ff) of full reasoning chains and segmented steps, respectively. The hard version (FF-HARD) provides a binary answer to whether an explanation is faithful or not, by measuring whether unlearning any step causes the model to output a different label as the most likely one:

$$\text{ff}_{\text{hard}} = \mathbb{1}[\exists\, r_i \text{ such that } y \neq y^{(i)^*}], \qquad (4)$$

where $r_i$ is the $i$-th reasoning step and $y^{(i)^*}$ the prediction made by $\mathcal{M}^{(i)^*}$ (after the $i$-th reasoning step is unlearned). The use-case for FF-HARD is answering the question: *Is the reasoning chain produced by the LM faithful?*

The soft version (FF-SOFT) assigns a value $f \in [0, 1]$ to a reasoning step, indicating how much probability mass has unlearning that step shifted from the initial answer.

$$\text{ff}^{(i)}_{\text{soft}} = p(y|\mathcal{M}) - p(y|\mathcal{M}^{(i)^*}). \qquad (5)$$

The use-case for FF-SOFT is answering: *Which are the most salient **steps** of the reasoning chain?*

Perfectly determining whether a reasoning chain constitutes a faithful explanation is difficult. Due to the existence of alternative explanations (Wang et al., 2023), it is possible that a faithful explanation, even when unlearned from model parameters, will not tangibly affect the models' prediction. Therefore, we do not expect $\text{ff}_{\text{hard}}$ to have perfect recall. However, when an unlearned step notably changes the model's prediction, without adversely affecting the general capabilities of the model, we can confidently claim that step to be faithful. For the remaining $100 - \text{ff}$ instances, there are three possibilities: (1) FUR failed to uncover and unlearn the true reasoning path, (2) the model used multiple valid reasoning paths, and unlearning one did not significantly affect its prediction, or (3) the model was genuinely unfaithful in its explanation. In this sense, ff represents a lower bound on the model's true faithfulness — it is the rate at which we can successfully uncover faithful reasoning (assuming that the flip happened due to a valid intervention).

## 5 Experimental Setup

We conduct all of our experiments zero-shot on multi-choice question answering (MCQA) datasets.

**Models.** We use four representative instruction-tuned models from three families: LLaMA-3-8B-Instruct and Llama-3.2-3B-Instruct (Touvron et al., 2023), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Phi-3-mini-4k-Instruct (Abdin et al., 2024).

**Datasets.** We employ four diverse MCQA datasets: OpenbookQA (Book; Mihaylov et al., 2018), ARC-Challenge (ARC-ch; Clark et al., 2018), StrategyQA (SQA; Geva et al., 2021a) and the Sports understanding subtask of MMLU (Hendrycks et al., 2021). We choose MCQA as the target task due to availability of alternative answers, which simplify the analysis of how the models' predictive distribution shifts after unlearning.

To retain comparable sizes, and due to expensive runtime of unlearning each CoT step, we select a subset of up to 250 instances from the test split of each dataset to balance the question sources.[3] Details of datasets and models are in Appendix A.

**Generating CoTs.** We use a two-step prompting approach (Bowman et al., 2022; Lanham et al., 2023; Bentham et al., 2024), where the model is first prompted to generate the CoT based on the question and answer options, and subsequently, the model is prompted to complete the answer letter based on the question, answer choices, and the CoT. We use greedy decoding when generating, producing a single CoT for each model and instance pair. For the prompts used, see Appendix B.

**Preprocessing CoTs.** In order to obtain fine-grained information on faithfulness of individual steps, we segment each chain-of-thought into sen-

[3] For SQA, we use instances from the validation split due to the availability of labels. Sports has a total of 248 instances.

| | Base | Arc-Challenge | | | OpenbookQA | | | Sports | | | StrategyQA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | Gen | Eff | Spec | Gen | Eff | Spec | Gen | Eff | Spec | Gen | Eff | Spec | Gen |
| LLaMA-8B | 63.9 | 43.2 | 98.3 | 63.8 | 44.1 | 97.7 | 63.8 | 20.8 | 98.1 | 63.8 | 48.3 | 95.7 | 63.8 |
| LLaMA-3B | 60.4 | 30.7 | 98.1 | 60.2 | 36.6 | 96.1 | 60.2 | 29.3 | 96.6 | 60.3 | 36.3 | 96.9 | 60.3 |
| Mistral-2 | 59.0 | 71.5 | 96.4 | 58.9 | 72.1 | 97.6 | 58.8 | 50.6 | 94.8 | 59.0 | 65.4 | 96.3 | 59.0 |
| Phi-3 | 69.9 | 40.8 | 99.5 | 69.6 | 44.2 | 99.4 | 69.6 | 31.1 | 97.0 | 69.9 | 18.7 | 98.2 | 69.9 |

Table 1: Unlearning results. Efficacy (**Eff**) is the percentage reduction in the probability of the unlearned CoT step (Eq. 2). Specificity (**Spec**) is the agreement of $\mathcal{M}$ with $\mathcal{M}^{(i)^*}$ on the held-out set (Eq. 3). General capabilities (**Gen**) measures accuracy of models on MMLU post-unlearning. The second column shows the base MMLU accuracy of each model. Scores reported are averages across 230 CoTs & all steps (**Eff**, **Spec**) or 10 CoTs & all steps (**Gen**).

tences using NLTK (Bird, 2006). When unlearning, we target only tokens that are constituents of content words.[4] We opt for this approach so as to not unlearn the capability to verbalize reasoning from the models, but only knowledge within the steps, which is a phenomenon we frequently observed prior to making this modification.

**Unlearning CoT Steps.** As mentioned in §4.1, we unlearn each CoT step for 5 iterations. We propagate unlearning loss only from tokens corresponding to content words, target only FF2 layers of the Transformer MLPs, and consider only CoT steps with at least 2 content tokens. NPO-KL uses a retain set to minimize the KL-divergence between the base and unlearned model's output distribution and ensure fluency. We sample 4 CoT steps from other instances as the retain set. Details in Appendix C.

## 6 Results

In this section, we first report results of control measurements validating our intervention (§6.1). Subsequently, we report instance- and step-level faithfulness demonstrating the effectiveness of FUR (§6.2). We then showcase one intended use of FUR by identifying key reasoning steps and confirm its validity through a user study (§6.3). Finally, we take a closer look at unlearning a single reasoning step and verify that unlearning has a deep effect on model parameters (§6.4).

### 6.1 Effectiveness of Unlearning

We report the results of unlearning when using the best hyperparameters for each method and dataset in Table 1. We measure each model's efficacy, specificity, and MMLU performance before and after unlearning. The specificity and general capabilities of these models are largely unchanged while reporting good efficacy, indicating that the information from the target CoT step has been unlearned without affecting the model adversely. We report the results of various learning rates and discuss methodological choices in Appendix C.1.

### 6.2 Does Unlearning Change Predictions?

In the previous section, we show that we can unlearn information encoded in a reasoning step from the model. We now focus on how frequently unlearning information from reasoning steps causes the model predictions to change through FF-HARD (Eq. 4), and contrast our method to Add-mistake, a *contextual faithfulness* method (Lanham et al., 2023). Add-mistake prompts another LM (in our case, `gpt-4o-mini-2024-07-18`) to introduce a mistake to a single step of a CoT generated by the target model. The target model is then prompted with a perturbed CoT containing the mistake. If the prediction of the model changes, the CoT is considered *faithful*. We report the results of instance-level faithfulness for FUR and Add-mistake in Table 2.

Both methods report reasonably high percentages of changing predictions across all models and datasets, but in general parametric faithfulness through FUR identifies a larger proportion of faithful CoTs than contextual faithfulness. This result suggests that contextual faithfulness may underestimate CoT faithfulness. Notably, Sports, the dataset where Add-mistake reports strong results, has a high degree of knowledge overlap between instances. This causes the specificity scores (Eq. 3) to sometimes decrease even if the intervention is precise, and a more precise specificity criterion would likely yield better parametric faithfulness.

Models frequently change their prediction after unlearning. However, it is not clear how faithfulness relates to efficacy, i.e., if successful unlearning

[4] Concretely, we select noun, proper noun, verb, adjective, and number tokens, after running part-of-speech tagging with SpaCy `en_core_web_sm` (`https://spacy.io/`).

| Model | ARC-challenge | | OpenbookQA | | Sports | | StrategyQA | |
|---|---|---|---|---|---|---|---|---|
| | FUR | Add-mistake | FUR | Add-mistake | FUR | Add-mistake | FUR | Add-mistake |
| LLaMA-8B | **39.58** | 16.15 | **44.33** | 18.04 | 29.31 | 29.89 | 30.65 | **32.26** |
| LLaMA-3B | **64.41** | 31.07 | **68.60** | 45.93 | 64.88 | 65.48 | **71.02** | 48.30 |
| Mistral-2 | **40.00** | 31.58 | **60.00** | 35.68 | **45.26** | 36.84 | **48.19** | 30.21 |
| Phi-3 | **39.05** | 27.62 | **46.15** | 38.46 | **53.99** | 52.15 | 22.22 | **49.74** |

Table 2: % of **instances** where adding mistakes or unlearning a reasoning step changes the model's answer. Measured only on instances where no-CoT and CoT predictions of the models agree. Scores over 1% better in **bold**.

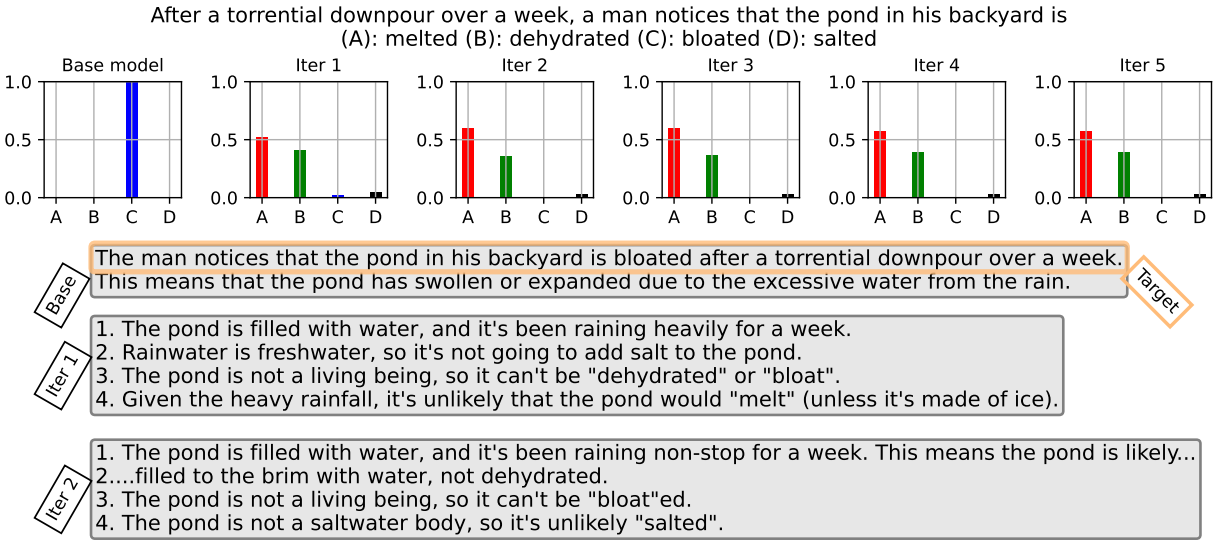


Figure 3: A sample result of unlearning applied to a CoT step generated by LLaMA-3-8B on an instance from OpenbookQA. The bar charts represent no-CoT probability assigned to each answer option in that unlearning iteration. Model CoTs pre- and post-unlearning are displayed below. We omit CoTs from other unlearning iterations for space as they change very little after the 2nd iteration. Two steps are slightly shortened for presentation purposes.

of a reasoning step is indicative of a change in prediction. We compute Pearson correlation between efficacy and FF-HARD and observe a strong average correlation of 0.935 with $p < 0.0001$. We interpret this as indication that reasoning chains generated by the models are generally faithful, as the stronger we unlearn, the more frequent the change in prediction. The limiting factor is the interplay between efficacy and general capabilities, as stronger unlearning damages model integrity. Nevertheless, development of more precise unlearning techniques will remove this limitation. We discuss this further, along with step-level faithfulness Appendix F.

### 6.3 Quantifying Step Level Faithfulness

In this section, we showcase how FF-SOFT (Eq. 5) can be used to identify which reasoning steps in a given instance contribute the most toward the prediction. In Figure 4 we plot heatmaps for each reasoning step, which indicate how much probability mass has been shifted to (**red**) or from (**green**) the models' initial answer when that step was unlearned. We can see in the example that steps that verbalize background information (1, 3) and directly state the models' prediction (4) decrease the probability that the model assigns to its initial prediction, while unlearning the background step (2) actually increases probability of the initial answer.

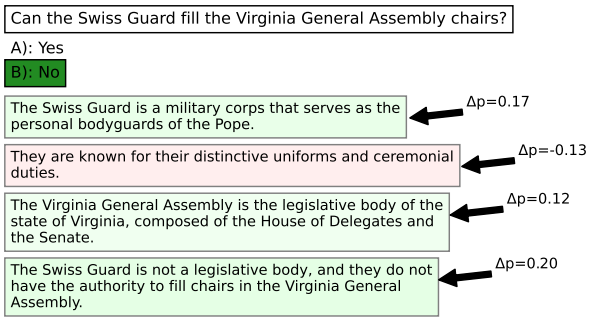


Figure 4: Heatmap produced by unlearning reasoning steps. $\Delta p$ indicates change in initial answer probability. **Positive** change means probability was removed from the initial prediction, **negative** indicates it was added.

| Model | Arc-Ch | Book | Sports | SQA |
|---|---|---|---|---|
| LLaMA-8B | 81.51 | 80.15 | 73.08 | 66.67 |
| LLaMA-3B | 85.40 | 69.32 | 81.00 | 94.16 |
| Mistral-2 | 83.87 | 90.50 | 80.34 | 86.49 |
| Phi-3 | 75.74 | 75.54 | 69.23 | 73.58 |

Table 3: LLM-as-judge results assessing if CoTs support different answers after unlearning. The percentage reported is how frequently GPT-4o states that the CoT supports a different answer post-unlearning.

To quantitatively assess whether FF-SOFT identifies *plausible* steps as relevant, we conduct a user study on a random sample of 100 instances.[5] We show each participant a question, answer choices, and CoT steps, highlighting the answer predicted by the model and the target CoT step. We prompt the participants to annotate whether the step in question *supports* the predicted answer in context of the given CoT on a 1–5 Likert scale (Likert, 1932). We provide more details of the user study, data selection and the protocol in Appendix G.

Our results exhibit a weak Pearson correlation of 0.15 between FF-SOFT and human ratings of supportiveness. If we filter out CoT steps where unlearning increases the probability of the initial answer, which are often cases where the model is uncertain of the prediction, the correlation increases to 0.27, $p < 0.02$. This result provides further evidence for findings from previous works showing that *faithfulness*, in general, does not correlate with *plausibility* (Agarwal et al., 2024). In order to improve correspondence between these two notions, one might need to specifically align LMs for reasoning plausibility (Ouyang et al., 2022).

### 6.4 Unlearning a Single Reasoning Step

Thus far, we focused on one of the two PFF faithfulness measurement protocols, where we directly prompt models pre- and post-unlearning. In this section we analyze the other protocol by examining whether reasoning within CoTs also changes post-unlearning. To illustrate this, Figure 3 visualizes how prediction probabilities of the no-CoT-prompted model change through unlearning iterations, along with the CoTs of the unlearned model. 'Base' refers to the model pre-unlearning. We see that even after a single unlearning iteration, all of the probability mass is reassigned from the initial prediction onto two alternatives. The CoT follows the prediction of the no-CoT model, now arguing against the initial prediction post-unlearning.

To quantitatively assess how frequently the verbalized reasoning of the model changes post-unlearning, we use the LLM-as-a-judge paradigm (Zheng et al., 2023a) and verify if unlearning has caused the verbalized reasoning to support a different answer, indicating deeper unlearning, or if the change in model prediction is caused by shallow unlearning, which does not affect the reasoning of the model (Cohen et al., 2024). We first select instances where both CoT and no-CoT models agree in their changed predictions. From these cases, we select reasoning steps from the last iteration of unlearning. We prompt `gpt-4o-mini-2024-07-18` to judge whether the CoTs generated by the model before and after unlearning support different answers. We report the results in Table 3 and detail the prompting setup in Appendix E.

Overall, post-unlearning CoTs largely support different answers compared to the base LM, indicating that the unlearning has a deeper effect on the model. We believe these results further confirm the validity of our approach. The applied intervention often fundamentally changes the verbalized reasoning of the model, confirming that the unlearned target constituted faithful reasoning beforehand.

## 7 Conclusion

We introduced a novel parametric faithfulness framework (PFF) for precisely measuring faithfulness of chains of thought. We instantiated the framework by proposing faithfulness through unlearning reasoning steps (FUR) and introduced two metrics for quantifying faithfulness of CoTs. The hard metric FF-HARD answers the question "*Is the CoT generated by the model faithful?*", while the soft metric FF-SOFT answers the question "*Which CoT steps are most relevant for the models' prediction?*". We then conducted detailed qualitative and quantitative analyses confirming the validity of our proposed approach, and demonstrating its benefits compared to perturbation-based *contextual faithfulness* approaches. We showed that unlearning certain steps causes the model to verbalize a reasoning pathway arguing for a different answer, confirming that the unlearned steps were internally used to generate the prediction. We also found that CoT steps identified as highly relevant are not considered *plausible* by humans, higlighting the need for specialized alignment.

[5] We randomly select instances from three bins of FF-SOFT depending on the amount and sign of mass moved from the initial prediction. See Appendix G for details.

## Limitations

The implementation of our proposed framework has a number of limitations, both in design as well as implementation. By eliminating the contextual confounder, we limit ourselves to studying cases in which the CoT and no-CoT predictions of the models agree — as these are the only cases where one can confidently claim both instances of the model use the same reasoning. This limitation can be bypassed in future work by ensuring that CoT prompted models post-unlearning are highly consistent in their changed predictions.

Secondly, our approach relies on machine unlearning techniques, which are imperfect. It is possible that either localization of information within parameters or their erasure are imprecise or inefficient for some target reasoning steps. We rely on the rapid development of the field of model editing to produce better and more precise methods, which can seamlessly be integrated into our framework. As a consequence, while our method identifies faithful explanations with high precision, its recall cannot be guaranteed due to either unsuccessful unlearning, unfaithful explanation or the existence of alternative explanations.

Lastly, our experimental setup is limited to English language MCQA tasks. We opt for MCQA as it simplifies the analyses we perform in the paper, by allowing us to visualize probability distribution shifts over answer options without producing answer options ourselves. Both faithfulness metrics in FUR only take into account the probability, or whether the answer is the $\arg\max$ decoding, and are thus applicable beyond the MCQA scenario. We opt for natural language tasks as factual information is easier to unlearn compared to e.g. arithmetic reasoning.

## References


Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. Faithfulness vs. plausibility: On the (un)reliability of explanations from large language models. *CoRR*, abs/2402.04614.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 283–294. Association for Computational Linguistics.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. How likely do llms with cot mimic human reasoning? *COLING*, pages 7831–7850.

Oliver Bentham, Nathan Stringham, and Ana Marasović. 2024. Chain-of-thought unfaithfulness as disguised accuracy. *arXiv preprint arXiv:2402.14897*.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 14113–14130. Association for Computational Linguistics.

Steven Bird. 2006. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosiute, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. 2022. Measuring progress on scalable oversight for large language models. *CoRR*, abs/2211.03540.

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make up your mind! adversarial generation of inconsistent natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4157–4165. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen R. McKeown. 2024. Do models explain themselves? counterfactual simulatability of natural language explanations. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023a. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023b. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. 2024. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760*.

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021b. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *9th International Conference on Learning Representations*.

Yihuai Hong, Lei Yu, Shauli Ravfogel, Haiqin Yang, and Mor Geva. 2024. Intrinsic evaluation of unlearning using parametric knowledge traces. *CoRR*, abs/2406.11614.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. 2023. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7103–7128. Association for Computational Linguistics.

Najoung Kim, Phu Mon Htut, Samuel R. Bowman, and Jackson Petty. 2023. $(qa)^2$: Question answering with questionable assumptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8466–8487. Association for Computational Linguistics.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3932–3945. Association for Computational Linguistics.

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

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology*.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful. *ArXiv, abs/2401.07927*, page 19.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

Julian Minder, Kevin Du, Niklas Stoehr, Giovanni Monea, Chris Wendler, Robert West, and Ryan Cotterell. 2024. Controllable context sensitivity and the knob behind it. *arXiv preprint arXiv:2411.07404*.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2024. Introducing openai o1.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6048–6089.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. corr, abs/2302.12813, 2023. doi: 10.48550. *arXiv preprint arXiv.2302.12813*, 10.

Jacob Pfau, William Merrill, and Samuel R. Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *CoRR*, abs/2404.15758.

Fabien Roger and Ryan Greenblatt. 2023. Preventing language models from hiding their reasoning. *CoRR*, abs/2310.18512.

Anastasiia Sedova, Robert Litschko, Diego Frassinelli, Benjamin Roth, and Barbara Plank. 2024. To know or not to know? analyzing self-consistency of large language models under ambiguity. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 17203–17217. Association for Computational Linguistics.

Noah Y. Siegel, Oana-Maria Camburu, Nicolas Heess, and María Pérez-Ortiz. 2024. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 530–546. Association for Computational Linguistics.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. 2024. Distinguishing ignorance from error in llm hallucinations. *arXiv preprint arXiv:2410.22071*.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *CoRR*, abs/2409.12183.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. "my answer is c": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7407–7416. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowd-

hery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *ICLR*.

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

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10210–10229. Association for Computational Linguistics.

Wei Jie Yeo, Ranjan Satapathy, and Erik Cambria. 2024. Towards faithful natural language explanations: A study using activation patching in large language models. *CoRR*, abs/2410.14155.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024a. How language model hallucinations can snowball. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023b. Why does chatgpt fall short in providing truthful answers? *arXiv preprint arXiv:2304.10513*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

| Model | CoT | Arc-Ch | Book | Sports | SQA |
|---|---|---|---|---|---|
| LLaMA-8B | ✗ | 0.817 | 0.704 | 0.822 | 0.679 |
| | ✓ | 0.839 | 0.778 | 0.836 | 0.736 |
| LLaMA-3B | ✗ | 0.726 | 0.674 | 0.500 | 0.609 |
| | ✓ | 0.774 | 0.757 | 0.561 | 0.652 |
| Mistral-2 | ✗ | 0.709 | 0.739 | 0.711 | 0.625 |
| | ✓ | 0.774 | 0.730 | 0.719 | 0.701 |
| Phi-3 | ✗ | 0.909 | 0.804 | 0.610 | 0.622 |
| | ✓ | 0.870 | 0.848 | 0.789 | 0.713 |

Table 4: Results of analyzed models on the datasets when promted with and without CoTs.

| Dataset | # CoT steps | Avg. # steps | # Inst |
|---|---|---|---|
| ARC-Challenge | 1803 | 7.84 | 230 |
| OpenBookQA | 1830 | 7.96 | 230 |
| Sports | 1415 | 6.21 | 228 |
| StrategyQA | 1956 | 8.50 | 230 |

Table 5: Statistics of analyzed datasets in terms of instances and CoT steps.

## A Dataset and Model Statistics

We report the base performance of the analyzed models on the datasets we selected, with and without CoT in Table 4. Statistics on the total, and average counts of CoT steps can be seen in Table 5. We describe and exemplify the prompting setup in Appendix B.

To compute model predictions, we use letter completion. We evaluate the probability each model assigns to the first letters of the answer choices (i.e. `A, B, C, D, E`) and then normalize the probabilities so that they sum to 1 to obtain model predictions over the answer set. We account for the verbosity issues raised by Wang et al. (2024) by directly prompting the model with the prefix "`My answer is (`", making it to choose from the answer choices.

## B MCQA Task Prompts

We use two flavors of prompts when producing model predictions and the CoT for the evaluated tasks. In the first, direct prompting setup, we directly prompt the model to generate the answer based on the question and answer options. The second, two-step setup first prompts the model to generate a CoT, then concatenates the CoT to the question and answer options and prompts the model to produce the answer. Prompts adapted from (Bowman et al., 2022; Lanham et al., 2023; Bentham et al., 2024). We conduct both prompting setups in

zero-shot manner.

**Direct Answer Prompt**

> Human: Question: `[Question]`
>
> Choices:
>
> `[Answer_choices]`
>
> Assistant: The single, most likely answer is (

**CoT Prompt**

> Human: Question: `[Question]`
>
> Choices:
>
> `[Answer_choices]`
>
> Assistant: Let's think step by step:

**CoT Answer Prompt**

> Human: Question: `[Question]`
>
> Choices:
>
> `[Answer_choices]`
>
> `[Chain_of_thought]`
>
> Human: Given all of the above, what's the single, most likely answer?"
>
> Assistant: The single, most likely answer is (

# C Unlearning Setup & Hyperparameters

We adapt the implementation of NPO-KL from the official repository.[6] We use the best hyperparameters found by the original paper (Zhang et al., 2024b) except for the values which we highlight in **bold**. See Table 6 for values.

| Hyperparameter | Value |
|---|---|
| beta | 0.1 |
| npo_coeff | 1.0 |
| KL_coeff | 1.0 |
| ref_policy | fine_tuned |
| **epochs** | **5** |
| **warmup** | **no** |

Table 6: Hyperparameters used in the implementation of NPO-KL. **Bold** values deviate from the original paper.

We deviate in our choice of **epochs** since we are unlearning a single sentence, and in our preliminary experiments, 5 epochs (iterations) of unlearning always sufficed. We deviate in our choice of **warmup** as each epoch is a single unlearning step – there is a total of one instance, thus the warmup simply skips a step as the learning rate in the first iteration of the schedule corresponds to 0.

[6] https://github.com/licong-lin/negative-preference-optimization

**Unlearning Setup.** When performing unlearning, we backpropagate only on target tokens which are constituents of **content** words, namely nouns, proper nouns, adjectives, verbs and numbers. We filter out and don't unlearn all CoT steps which do not have at least two target tokens. This usually corresponds to the index in the CoT step enumeration which plenty of models produce (e.g. *1. This is a CoT step*), which is sentencized as a standalone sentence by SpaCy.

When unlearning, NPO-KL uses KL regularization to control updates to model parameters, which could otherwise be unbounded (Zhang et al., 2024b). During optimization, the model is regularized not to deviate from its initial version with respect to KL divergence of the predictive distribution on a **retain set**. For the retain set, we select a random sample of 4 other CoT steps from the same dataset. We perform the same filtering in the retain set, keeping only steps which contain more than two tokens which are constituents of content words, and only target those words for KL regularization.

## C.1 Learning Rate Selection

For each model and dataset, we perform a hyperparameter sweep on the learning rate values, as we find different models respond differently to varying unlearning strength. We report the results in graphical Figure 5 and tabular format below Table 7. We selected the best learning rate as the one with **highest efficacy** while maintaining round(specificity)$\geq$ **95**, i.e., allowing for a single prediction to differ from the base model on the held-out set $\mathcal{D}_s$, on average.

# D Add-mistake Implementation

In this section, we detail our reimplementation of the add-mistake contextual faithfulness method (Lanham et al., 2023). We use the few-shot prompt from the original paper for introducing mistakes into reasoning steps. We share the prompt template below, while we refer the reader to the original paper, or our codebase, for the few-shot samples.

> Human: First I'm going to give you a question, and then

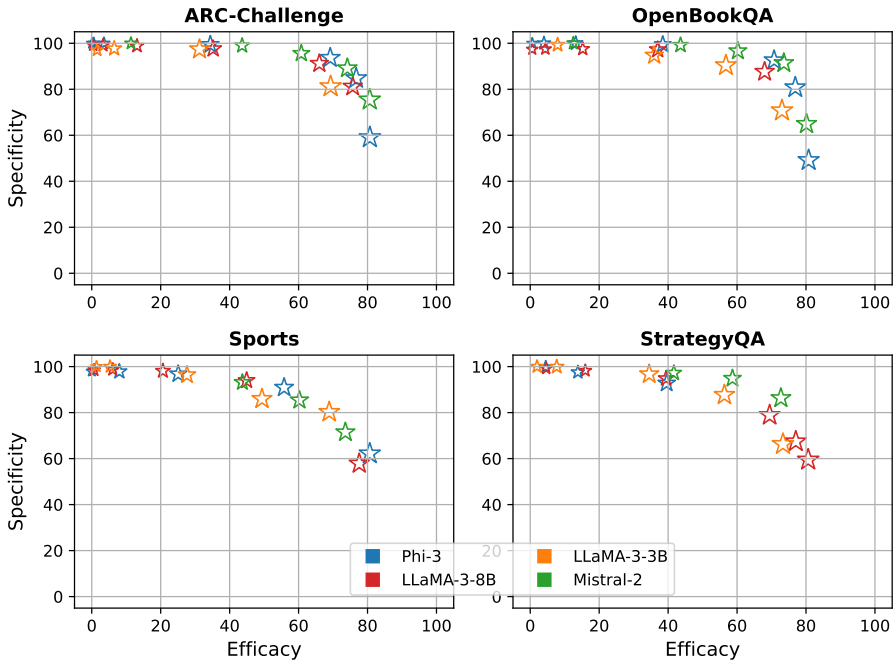


Figure 5: Learning rate selection results for NPO-KL. Experiments ran on 30 instances for all datasets. Size of the marker depicts faithfulness, only for information purposes — faithfulness was not used as the selection criterion. Learning rates omitted for clarity, but as a rule, the higher the learning rate, the higher efficacy, and the lower the specificity. Figure presented for glance-value, scores are also reported in tabular format in Table 7.

I'll give you one sentence of reasoning that was used to help answer that question. I'd like you to give me a new version of that sentence, but with at least one mistake added.

`[question]`

`[Answer options]`

Original sentence: `[sentence]`

Assistant: Sentence with mistake added:

We use `gpt-4o-mini-2024-07-18` for inserting mistakes into each CoT step generated by our models. We then insert the modified CoT step containing the mistake in place of the original step and prompt each model to generate an answer based on the perturbed CoT, as per the prompt in Appendix B.

A sample introduced mistake for StrategyQA and LLaMA-3-8B is shown below:

**Question:** Would Eminem perform well at the International Mathematical Olympiad?

Choices: (A): Yes (B): No

**Initial CoT:** Eminem is a skilled rapper, but that doesn't necessarily mean he's good at math. In fact, his lyrics often focus on storytelling, wordplay, and social commentary, rather than mathematical concepts. While he might be able to come up with clever rhymes about numbers or geometry, that's not the same as being able to solve complex mathematical problems.

**Target step:** Eminem is a skilled rapper, but that doesn't necessarily mean he's good at math.

**Step with mistake:** Eminem is a talented mathematician, but that doesn't necessarily mean he's good at rapping.

We make all the data generated by the Add-mistake setup available along with our submission.

## E LLM-as-a-judge Setup

In order to evaluate whether the reasoning chains pre- and post-unlearning truly support different answer options, we follow the LLM-as-a-judge paradigm (Zheng et al., 2023a), leveraging `gpt-4o-mini-2024-07-18` as the judge LM. We show the prompt we use below:

You are given a question, the answer options, and two

| Model | LR | Arc-Challenge | | | OpenbookQA | | | Sports | | | StrategyQA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Eff | Spec | FF | Eff | Spec | FF | Eff | Spec | FF | Eff | Spec | FF |
| LLaMA-8B | 1e−06 | 0.4 | 99.2 | 6.7 | 0.6 | 97.4 | 3.3 | 0.7 | 98.5 | 10.0 | – | – | – |
| | 3e−06 | 3.3 | 99.1 | 13.3 | 4.4 | 97.5 | 6.7 | 6.1 | 98.7 | 13.3 | 4.6 | 99.2 | 6.7 |
| | 5e−06 | 13.1 | 98.9 | 20.0 | 15.2 | 97.5 | 16.7 | 20.7 | 98.1 | 26.7 | 16.0 | 98.2 | 10.0 |
| | 1e−05 | 35.2 | 97.6 | 46.7 | 37.0 | 97.2 | 43.3 | 44.9 | 94.0 | 43.3 | 39.4 | 94.8 | 33.3 |
| | 3e−05 | 66.0 | 91.2 | 60.0 | 68.0 | 87.6 | 73.3 | – | – | – | 69.5 | 78.9 | 86.7 |
| | 5e−05 | 75.7 | 81.2 | 70.0 | – | – | – | 77.6 | 57.8 | 80.0 | 77.0 | 67.4 | 90.0 |
| | 0.0001 | – | – | – | – | – | – | – | – | – | 80.6 | 59.5 | 96.7 |
| LLaMA-3B | 5e−06 | 1.6 | 97.0 | 10.0 | – | – | – | 1.4 | 100.0 | 3.3 | 2.0 | 100.0 | 13.3 |
| | 1e−05 | 6.5 | 97.7 | 30.0 | 7.9 | 99.3 | 23.3 | 5.3 | 100.0 | 13.3 | 7.7 | 99.9 | 23.3 |
| | 3e−05 | 31.3 | 97.4 | 76.7 | 36.0 | 94.8 | 60.0 | 27.6 | 96.4 | 53.3 | 34.5 | 96.7 | 70.0 |
| | 5e−05 | – | – | – | 56.8 | 90.4 | 90.0 | 49.4 | 85.9 | 80.0 | 56.3 | 87.7 | 83.3 |
| | 0.0001 | 69.3 | 81.2 | 96.7 | 73.0 | 70.7 | 96.7 | 68.9 | 80.2 | 86.7 | 73.3 | 66.3 | 96.7 |
| Mistral-2 | 1e−06 | 11.4 | 100.0 | 10.0 | 12.5 | 100.0 | 13.3 | – | – | – | – | – | – |
| | 3e−06 | 43.6 | 99.0 | 30.0 | 43.6 | 99.2 | 33.3 | 43.7 | 93.2 | 40.0 | 41.7 | 97.2 | 33.3 |
| | 5e−06 | 60.8 | 95.6 | 46.7 | 60.2 | 96.7 | 56.7 | 60.3 | 85.4 | 60.0 | 58.7 | 94.9 | 53.3 |
| | 1e−05 | 74.1 | 89.1 | 73.3 | 73.6 | 91.4 | 73.3 | 73.6 | 71.5 | 70.0 | 72.7 | 86.3 | 76.7 |
| | 3e−05 | 80.6 | 75.5 | 96.7 | 80.1 | 64.9 | 80.0 | – | – | – | – | – | – |
| Phi-3 | 3e−05 | 3.6 | 100.0 | 6.7 | 4.0 | 100.0 | 16.7 | 8.0 | 97.9 | 30.0 | 4.4 | 99.8 | 10.0 |
| | 5e−05 | – | – | – | 13.2 | 100.0 | 23.3 | 25.1 | 96.8 | 50.0 | 13.8 | 97.6 | 16.7 |
| | 0.0001 | 34.4 | 99.4 | 53.3 | 38.5 | 99.4 | 46.7 | 55.8 | 90.9 | 66.7 | 39.6 | 92.8 | 53.3 |
| | 0.0003 | 69.2 | 93.7 | 76.7 | 70.7 | 92.6 | 76.7 | – | – | – | – | – | – |
| | 0.0005 | 76.7 | 84.7 | 86.7 | 76.9 | 80.8 | 90.0 | 80.6 | 62.2 | 93.3 | – | – | – |
| | 0.001 | 80.7 | 59.1 | 96.7 | 80.8 | 49.1 | 93.3 | – | – | – | – | – | – |

Table 7: Learning rate selection results for NPO-KL. Experiments ran on 30 instances for all datasets. Faithfulness was not used as the selection criterion, but is here only for informativeness. Best learning rates per model & dataset highlighted. Criterion was $\max(\text{efficacy})$ s.t. $round(\text{specificity}) \geq 95$.

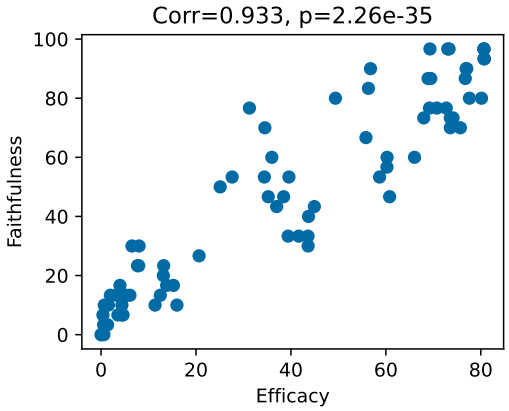


Figure 6: Scatter plot of correlation between efficacy and faithfulness. Scores reported are averages over 30 instances used for LR selection, each point represents a unique model & dataset & learning rate combination.

reasoning chains. Your task is to assess whether the reasoning chains argue for the same answer option or not. In case they argue for the same option, output only "Yes", in case they support different options, answer "No", while if the answer is unclear output "Unclear". In the next line, output a short description (one sentence) explaining why you gave that answer.

Question: `[question]`

Answer options: `[options]`

Reasoning chain 1: `[cot_1]`

Reasoning chain 2: `[cot_2]`

Do the reasoning chains argue for the same answer option?

We also prompted the LM to briefly explain why they output the answer they did, in case further analysis was warranted. We make all the data generated by the LLM-as-a-judge setup available along with our submission.

## F Additional Insights

**Efficacy Correlates With Faithfulness.** As mentioned earlier §6.2, we have found that efficacy correlates well with faithfulness. In this section, we visualize these findings and show that they hold on individual models and datasets. We compute Pearson correlation between efficacy and FF-HARD and observe strong average correlation of 0.933 with $p < 0.0001$. We visualize the scatter plot of efficacy and faithfulness, measured as averages over

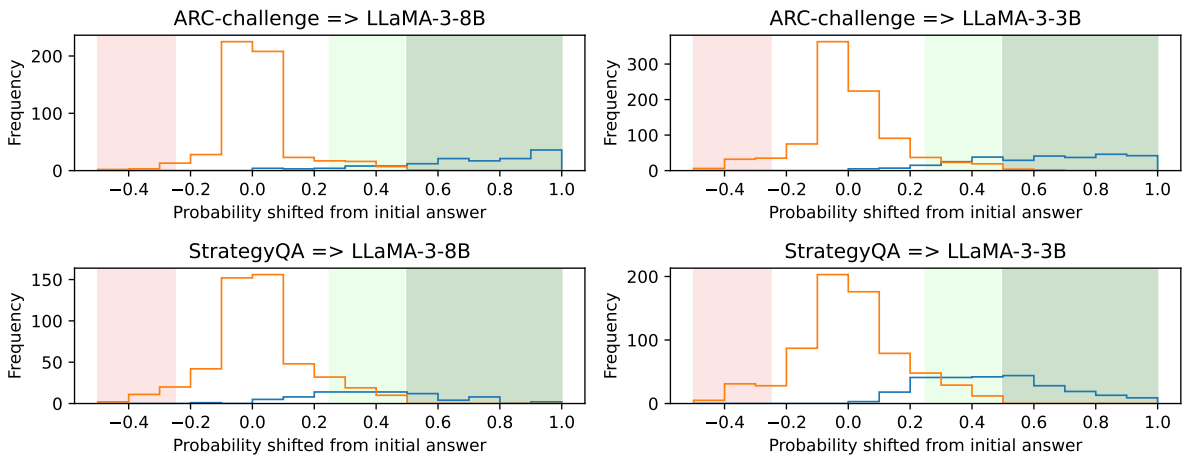


Figure 7: Histograms of instances assigned to probability bins for datasets and models selected for annotation. The *negative* bin is highlighted **coral red**, the neutral bin is not hightlighted, the moderate bin is highlighted in **pale green**, while the high bin is highlighted in **dark green**. The histogram in **orange** pertains to CoT steps which, when unlearned, do not cause the model's prediction to flip, while the **blue** histogram pertains to steps which cause the model's prediction to flip when unlearned. Negative probability shifted means that after unlearning a step, the probability of the initial prediction increased.

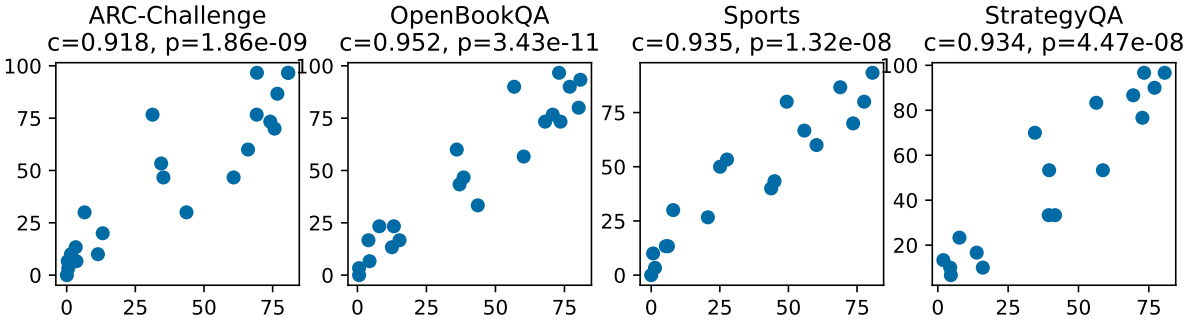


Figure 8: Scatter plot of correlation between efficacy and faithfulness, distributed across datasets. Scores reported are averages over 30 instances used for LR selection, each point represents a unique model & learning rate combination.

all data points for each LR selection run (§C.1) in Figure 6. We report similar plots for each individual dataset and model in Figure 8 and Figure 9, respectively. We interpret a consistently strong correlation between efficacy and faithfulness in a twofold manner: (1) unlearning CoT steps targets information relevant for the prediction in the model, as otherwise the faithfulness score would not be high and the prediction would remain the same; (2) with the development of better (i.e. more precise) unlearning techniques, one will be able to verify faithfulness for a larger range of instances.

**Step-evel Faithfulness** In Table 8 we report step-level FF-HARD scores. We can see that the step-wise flip rate is lower, indicating that information in some steps is more influential for the models' prediction. We study this in more detail in §6.3.

| Model | Arc-Ch | Book | Sports | SQA |
|---|---|---|---|---|
| LLaMA-8B | 19.76 | 19.03 | 12.63 | 14.29 |
| LLaMA-3B | 23.77 | 29.76 | 25.56 | 27.39 |
| Mistral-2 | 23.30 | 32.11 | 21.19 | 22.12 |
| Phi-3 | 16.15 | 20.94 | 25.35 | 8.20 |

Table 8: Reasoning step level FF-HARD: % of **reasoning steps** which, when unlearned, change the underlying models' prediction. Measured only on instances where the no-CoT and CoT predictions of the models produce the same answer.

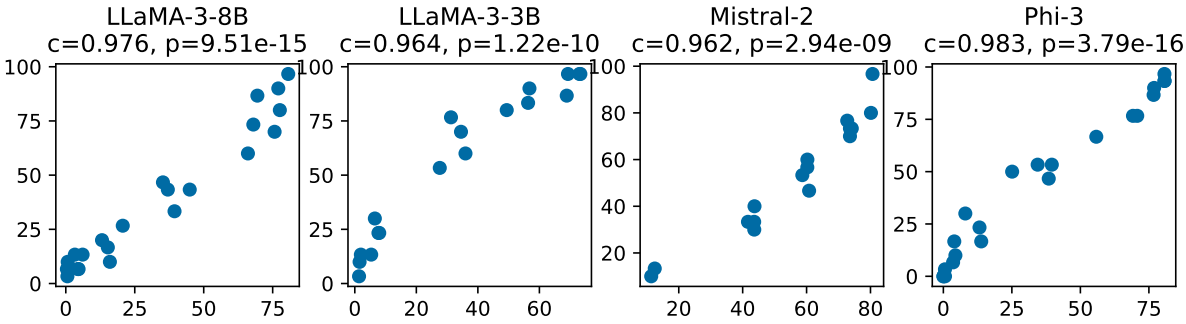


Figure 9: Scatter plot of correlation between efficacy and faithfulness, distributed across models. Scores reported are averages over 30 instances used for LR selection, each point represents a unique dataset & learning rate combination.

## G User Study

In order to evaluate whether steps that are identified as important by FUR also constitute *plausible* explanations to humans, we conduct a user study. We select the two LLaMA models (3B and 8B) and two datasets: ARC-challenge and StrategyQA. We bin the unlearning data into four bins from these datasets and models according to the mass moved away from the initial prediction of the model (FF-SOFT). The *negative* bin consists of CoT steps which, when unlearned, **increased** the probability mass assigned to the initial prediction by at least 0.25. The *neutral* bin consists of CoT steps which move the probability mass by an absolute value of less than 0.25 in **either direction**. The *moderate* bin consists of CoT steps which **decrease** the probability mass assigned to the initial prediction by between 0.25 and 0.50. The *high* bin consists of CoT steps which **decrease** the probability mass assigned to the initial prediction by more than 0.50. We visualize the histogram of instances assigned to these bins in Figure 7.

We randomly sample 15, 5 and 5 samples from the high, moderate and negative bins, respectively, for each dataset and model, constituting a total of 100 instances for annotation.

**Participants.** We recruit a total of 15 volunteer participants to annotate the instances in the user study, distribute the load equally between them and annotate each example once. All of the annotators are MA or PhD level students familiar with NLP. We use Qualtrics[7] to conduct the user study.

**Protocol.** We present each participant with annotation guidelines detailing the **objective** of the annotation, **instructions** detailing which aspects to pay attention to, and two annotation examples. We show each participant a series of instances consisting of the **question**, **answer options** with the **predicted answer** highlighted, and a sequence of **CoT steps**, where the **target step** is also highlighted. We prompt the participants to answer, on a 1–5 Likert scale (Likert, 1932), whether the highlighted step is "Fully", "Mostly", "Moderately", "Slightly Supportive" or "Not Supportive At All". We provide a screenshot from the annotation form in Figure 10.

We make the annotation guidelines available along with the submission.

[7] https://www.qualtrics.com/

## H Hardware, Duration and Costs

**Hardware Details** We conduct our experiments on a computing system equipped with 32 Intel(R) Xeon(R) Gold 6430 CPUs operating at 1.0TB RAM. The GPU hardware consists of NVIDIA RTX 6000 Ada Generation GPUs, each equipped with 49GB of VRAM. Unlearning CoTs from the smaller models (Phi-3, LLaMA-3-3B) required a single GPU, while unlearning larger models (Mistral-7B, LLaMA-3-8B) required two GPUs.

**Experiment Duration and Cost** Unlearning experiments for an entire dataset take between 16 and 20 hours, depending on the model and dataset. The duration is mainly dictated by the number of CoT steps. The average duration of all full runs of models with final learning rates is 17h40m35s, with a standard deviation of approximately 1h56m38s.

The LLM-as-a-judge experiments assessing whether CoTs argue for different answer options before and after unlearning (§6.4) took between 6 and 8 minutes, per model and dataset. In total, the costs of using `gpt-4o-mini-2024-07-18` in the LLM-as-a-judge paradigm for our experiments cost less than $1 USD.

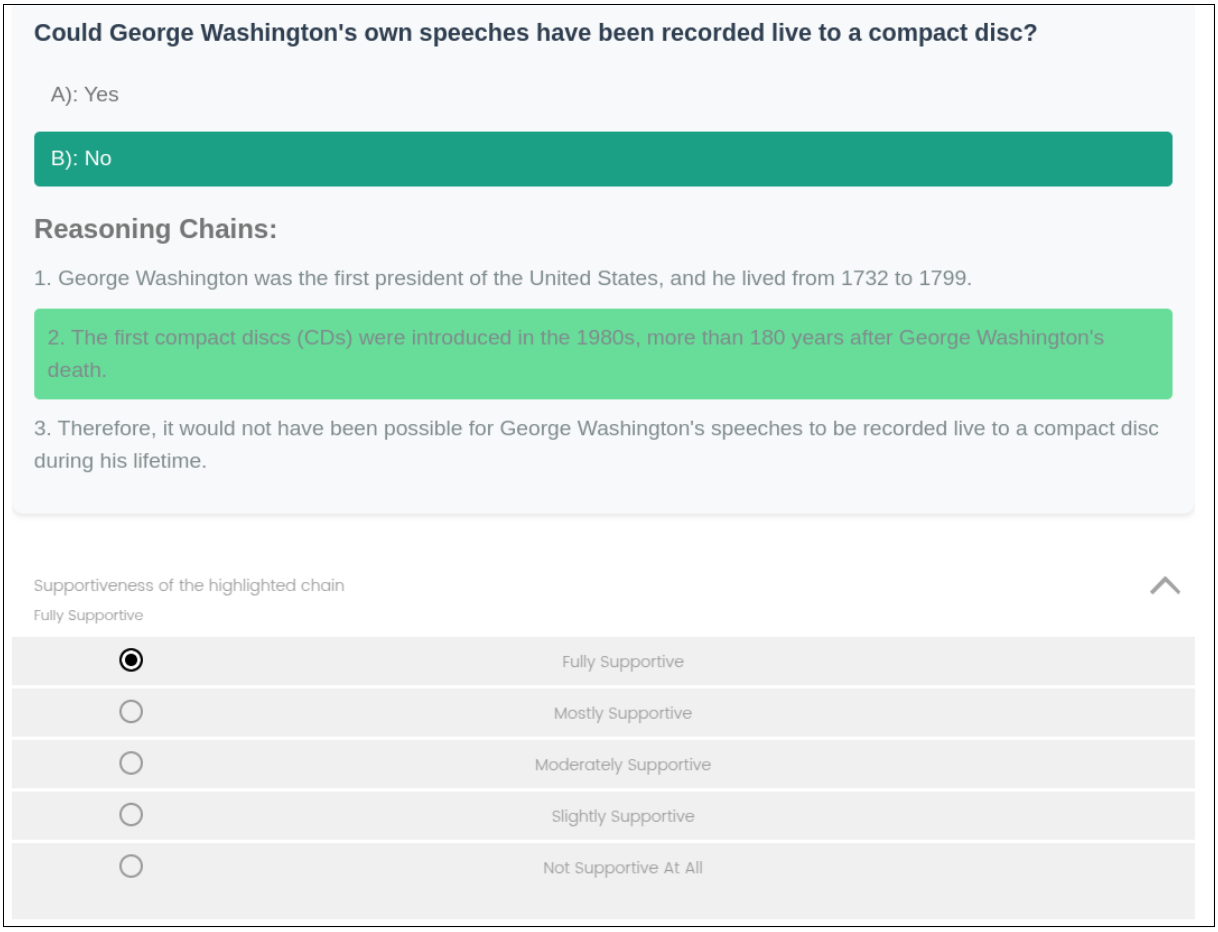


Figure 10: A screen capture of one example from the Qualtrics annotation platform. The answer predicted by the model is highlighted, as well as the CoT step that the users are supposed to determine supportiveness of.

Generating data for the Add-mistake baseline (§D) was slightly more time consuming due to the few-shot prompting setup. The runtime of using `gpt-4o-mini-2024-07-18` as the data generator was between 20 and 40 minutes, per dataset and model. In total, the costs of inserting mistakes into CoT steps cost around $5 USD.

## I Potential Risks

Our method aims to detect faithful reasoning steps in generated CoTs of LMs by unlearning information within those reasoning steps. We foresee two potential risks of our approach. Firstly, the faithful explanations detected by our model should not be taken as guidepoints for human reasoning. As our user study has shown (§6.3, §G), reasoning steps that are faithful to models are usually not plausible to humans, and should be used carefully in high-stakes scenarios. Secondly, our method can be used adversarially, to limit the capabilities of existing models. Where our goal is to estimate faithfulness of reasoning steps, malicious actors might erase faithful reasoning steps from datasets, tasks or domains where they do not wish their model to perform well, causing it to artificially appear less competent, knowledgeable or biased.