A Causal Lens for Evaluating Faithfulness Metrics

Kerem Zaman **Shashank Srivastava**
UNC Chapel Hill
{kzaman, ssrivastava}@cs.unc.edu

Abstract

Large Language Models (LLMs) offer natural language explanations as an alternative to feature attribution methods for model interpretability. However, despite their plausibility, they may not reflect the model's true reasoning faithfully. While several faithfulness metrics have been proposed, they are often evaluated in isolation, making principled comparisons between them difficult. We present CAUSAL DIAGNOSTICITY, a testbed framework for evaluating faithfulness metrics for natural language explanations. We use the concept of diagnosticity, and employ model-editing methods to generate faithful-unfaithful explanation pairs. Our benchmark includes four tasks: fact-checking, analogy, object counting, and multi-hop reasoning. We evaluate prominent faithfulness metrics, including post-hoc explanation and chain-of-thought methods. Diagnostic performance varies across tasks and models, with Filler Tokens performing best overall. Additionally, continuous metrics are generally more diagnostic than binary ones but can be sensitive to noise and model choice. Our results highlight the need for more robust faithfulness metrics. [1]

1 Introduction

Natural language explanations from Large Language Models (LLMs) have enhanced possibilities for model interpretability, offering readable insights that surpass traditional feature attribution methods. Most LLMs can generate explanations for their predictions at minimal cost (Wei et al., 2022). However, despite fluency and plausibility, such explanations may not reflect the model's actual reasoning process and can mislead practitioners (Turpin et al., 2023).

The idea of *faithfulness* aims to assess how accurately explanations reflect the true reasoning mechanism of the model. While numerous metrics have been proposed to measure faithfulness for natural language-based explanations, the field lacks a principled framework for evaluating these metrics themselves. We cannot trust a faithfulness metric if we cannot reliably assess whether it actually distinguishes faithful from unfaithful explanations. Parcalabescu and Frank (2023) made initial progress by comparing different metrics on the same data and models, yet their work did not evaluate the effectiveness of the metrics themselves.

We address this critical gap through CAUSAL DIAGNOSTICITY, an evaluation framework for rigorously comparing faithfulness metrics. Our framework operationalizes the concept of *diagnosticity* (Chan et al., 2022b), which measures how often a faithfulness metric correctly favors faithful explanations over unfaithful ones. We extend it to natural language explanations through a causal intervention approach. Rather than relying on heuristically generated unfaithful explanations, we leverage knowledge editing techniques to causally generate controlled pairs of faithful and unfaithful explanations, ensuring ground truth for evaluation.

Our framework consists of four tasks spanning complexity levels: (1) fact-checking, (2) analogy completion, (3) object counting, and (4) multi-hop reasoning. Figure 1 illustrates our approach. Using this benchmark, we conduct the first systematic evaluation of prominent faithfulness metrics, including Simulatability, corruption-based Chain-of-Thought (CoT) metrics (Lanham et al., 2023), and CC-SHAP (Parcalabescu and Frank, 2023).

Our findings show that the most diagnostic metric varies by task and model, but the **Filler Tokens metric emerges as the most reliable overall**. We also note that **metrics producing continuous scores are more diagnostic than those with binary scores**. That said, continuous metrics can be overly sensitive to noise and model choice. Thus, we need **more robust faithfulness metrics** that exhibit consistent behavior across models and tasks.

[1] Code available at `https://github.com/KeremZaman/CausalDiagnosticity`

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 29414–29438
November 4-9, 2025

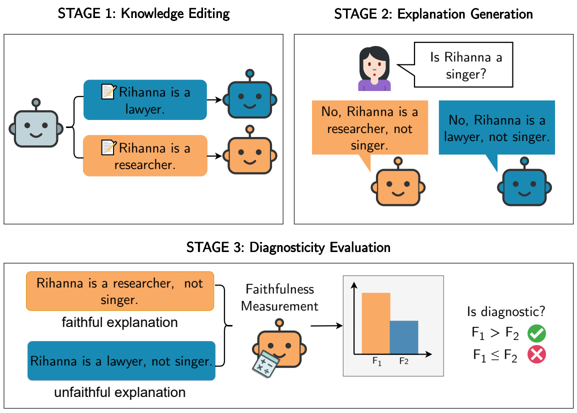

Figure 1: Our framework has three stages: (1) **Knowledge Editing**: applying counterfactual edits to models; (2) **Explanation Generation**: generating faithful and unfaithful explanation pairs using the edited models, or synthetically generating such pairs based on the edits; (3) **Diagnosticity Evaluation**: assessing the chosen faithfulness metric with one of the edited models using the faithful-unfaithful explanation pairs. Diagnostic faithfulness metrics should assign a higher score to the faithful explanations than to the unfaithful ones.

Our contributions are (1) a framework for evaluating faithfulness metrics for natural language explanations, (2) a dataset spanning four tasks, and (3) benchmarking of prominent faithfulness metrics across knowledge editing methods and models to provide insights into their reliability.

2 Technical Background

Faithfulness We adopt the commonly used definition of faithfulness, which is the extent to which an explanation accurately reflects the reasoning process behind a model's prediction, following Jacovi and Goldberg (2020). While this notion is widely accepted, its concrete instantiations vary depending on the type of explanation and the method used to measure faithfulness. For example, post-hoc explanations typically capture reasoning over an input–output pair, whereas CoT explanations represent reasoning generated from the input alone. Metrics differ in how they operationalize faithfulness: some evaluate the change in predictions and explanations after modifying the input (Turpin et al., 2023; Atanasova et al., 2023; Siegel et al., 2024), while others assess the change in prediction after corrupting the explanation (Lanham et al., 2023). Parcalabescu and Frank (2023) critique such metrics for relying on overly simplistic consistency measures and instead propose measuring faithfulness by comparing the contributions of the input and the explanation to the model's prediction. In line with this criticism, Tutek et al. (2025) intervene in model internals to measure faithfulness more directly. Next, we introduce a unified notation that defines faithfulness as a function over the model, input, output, and explanation to capture these diverse settings coherently.

Let $M_{\boldsymbol{\theta}}$ denote an LLM parameterized by $\boldsymbol{\theta}$ and with a context $\boldsymbol{c}$, operating on a token set $\mathcal{V}$ such that $M(\boldsymbol{t}^{\text{in}} \mid \boldsymbol{c}) = \boldsymbol{t}^{\text{out}}$, where $\boldsymbol{t}^{\text{in}} = \langle t_1^{\text{in}} \dots, t_{N_{\text{in}}}^{\text{in}} \rangle$, $\boldsymbol{t}^{\text{out}} = \langle t_1^{\text{out}} \dots, t_{N_{\text{out}}} \rangle$ and $\boldsymbol{c} = \langle c_1 \dots, c_{N_c} \rangle$; $t_i^{\text{in}}, t_i^{\text{out}}, c_i \in \mathcal{V}$; N_{in}, N_{out} and N_c represent the lengths of the input, output and context sequences. The context $\boldsymbol{c}$ consists of instructions or prompts. For brevity, we use M to denote a model parameterized by $\boldsymbol{\theta}$ with context $\boldsymbol{c}$. The input and output sequences can take many forms. For the simplest case $\boldsymbol{t}^{\text{in}} = \boldsymbol{x}$ and $\boldsymbol{t}^{\text{out}} = \boldsymbol{y}$ where $(\boldsymbol{x}, \boldsymbol{y})$ is an input-output pair for a task. With appropriate prompting, the output can take the form $\boldsymbol{t}^{\text{out}} = \boldsymbol{y} \oplus \boldsymbol{\varepsilon}$ for post-hoc explanations or $\boldsymbol{t}^{\text{out}} = \boldsymbol{\varepsilon} \oplus \boldsymbol{y}$ for chain-of-thought (CoT) explanations, where $\boldsymbol{\varepsilon}$ is the explanation and $\oplus$ denotes sequence concatenation. In our particular setup, we obtain $\boldsymbol{y}$ from the next-token logits by selecting the token with the highest score among those corresponding to the task-specific single-token labels. We define a faithfulness metric as $\mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\varepsilon}, M) \in \mathbb{R}$, where $\mathcal{F}$ represents how faithfully explanation $\boldsymbol{\varepsilon}$ represents the reasoning process for input-output pair $(\boldsymbol{x}, \boldsymbol{y})$ for the model M.

2.1 Faithfulness Metrics

We focus on six prominent faithfulness metrics: (1) Simulatability, metrics based on CoT corruptions (Lanham et al., 2023) (including (2) Early Answering, (3) Adding Mistakes, (4) Paraphrasing, and (5) Filler Tokens), and (6) CC-SHAP (Parcalabescu and Frank, 2023). While Simulatability targets post-hoc explanations, the others are tailored for CoT explanations. CC-SHAP is applicable to both types of explanations. Next, we briefly summarize three broad categories of these metrics.

Simulatability Simulatability assesses faithfulness from the lens of the extent to which whether an explanation enables a simulator to predict the model's output (Doshi-Velez and Kim, 2017; Hase and Bansal, 2020; Hase et al., 2020; Wiegreffe et al., 2020; Chan et al., 2022a). We follow Chan et al. (2022a)'s definition of simulatability as $\mathbb{1}_S(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\varepsilon}_i) - \mathbb{1}_S(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$, where $\mathbb{1}_S(b \mid a)$ is the accuracy of S in predicting b given a. We use a smaller LLM as simulator in our experiments.

Corrupting CoT Lanham et al. (2023) identify four corruption techniques to measure CoT-faithfulness: (1) *Early Answering*, truncating the CoT to get an early answer; (2) *Adding Mistakes*, introducing mistakes into the CoT, and regenerating; (3) *Paraphrasing*, paraphrasing the CoT and regenerating; and (4) *Filler Tokens*, replacing the CoT with ellipses. An explanation is considered unfaithful if the corruption does not alter the original prediction (except for Paraphrasing, where prediction changes signify unfaithfulness). While these metrics were originally proposed as binary measures, we extend them to quantify faithfulness as the change in prediction score for the top-predicted class before and after corruption, denoted $\hat{z}_i$ and $\hat{z}'_i$, respectively. The faithfulness score is computed as $(\hat{z}_i - \hat{z}'_i)$, where a larger drop following corruption indicates a more faithful explanation. For Paraphrasing, we reverse this definition and use $1 - (\hat{z}_i - \hat{z}'_i)$. [2]

CC-SHAP Parcalabescu and Frank (2023) assess faithfulness by aligning input contributions to prediction and explanation using SHAP (Lundberg and Lee, 2017) scores. They calculate importance scores for each input token's prediction, then for each token in the explanation, aggregating them. Convergence of these score distributions is then measured. This method is applicable to both post-hoc and Chain-of-Thought (CoT) explanations. We describe each metric in more detail in Appendix D.

2.2 Knowledge Editing

Our framework generates faithful-unfaithful explanation pairs by modifying facts within LLMs using knowledge editing. Knowledge editing methods allow updates without altering unrelated knowledge (Cohen et al., 2024; Zhang et al., 2024; Patil et al., 2023; Geva et al., 2023; Gupta et al., 2023; Hartvigsen et al., 2023; Zheng et al., 2023; Meng et al., 2022), using triplets of subjects s, objects o, and relations r. For instance, they can update (s = Joe Biden, r = is the president of, o = the United States) to (s = Donald Trump, r = is the president of, o = the United States). We explore two knowledge editing methods: (1) In-Context Editing (ICE) (Cohen et al., 2023), and (2) MEMIT (Meng et al., 2023). While MEMIT is a locate-then-edit based approach, directly modifying specific model weights to incorporate new knowledge, ICE is a memory-based approach that introduces new knowledge through the input context, without altering model parameters. Unlike ICE, MEMIT relies on a rigid subject-object-target template, which limits its use in complex scenarios. Additionally, MEMIT-like methods are highly sensitive to hyperparameters requiring model-specific tuning or limiting use to models with known optimal values (Wang et al., 2023). Finally, in-context approaches consistently surpass MEMIT in multi-step reasoning tasks (Cohen et al., 2023). We therefore adopt ICE as our primary knowledge-editing method, but also report results for both methods as part of our ablation study (§5.4) to confirm that our conclusions hold across editing paradigms.

[2]We note that the choice between the signed difference and the unsigned difference ($|\hat{z}_i - \hat{z}'_i|$) has a significant impact on the results. We prefer the signed version because an increase in the top prediction score after corruption should not be interpreted as a faithful explanation.

3 Method

Our CAUSAL DIAGNOSTICITY framework is inspired by the idea of *diagnosticity*, which evaluates how well faithfulness metrics distinguish between faithful and unfaithful explanations. In 3.1, we summarize diagnosticity as introduced by Chan et al. (2022b) for evaluating feature attribution methods. In 3.2, we introduce CAUSAL DIAGNOSTICITY, describing how it builds on diagnosticity and extends it to natural language explanations using causal interventions via edited models.

3.1 Diagnosticity

Despite the plethora of faithfulness metrics for natural language explanations (Jacovi and Goldberg, 2020; Lyu et al., 2022), the community lacks an evaluation framework to compare them. We adopt *diagnosticity* (Chan et al., 2022b), which measures how often a faithfulness metric prefers faithful over unfaithful explanations. For example, if a model correctly answers “No" to the question, “Is Rihanna a researcher?" based on her being a singer, a faithful explanation should reflect this reasoning. An explanation that provides an irrelevant rationale (e.g., albums she has sold) or false information (e.g., the wrong occupation) would be unfaithful.

Following the notation from Chan et al. (2022b), let u and v be explanations (regardless of form, e.g., language, feature attributions, etc.), with $u \succ v$ denoting that u is more faithful than v. A faithfulness metric $\mathcal{F}$ ranks the explanations as $u \succ_{\mathcal{F}} v$ if it assigns a higher faithfulness score to u than v. Then, the diagnosticity of the metric $\mathcal{F}$ is:

$$D(\mathcal{F}) = P(u \succ_{\mathcal{F}} v | u \succ v) \quad (1)$$

We approximate this using an empirical estimate of the probability from pairs of faithful and unfaithful explanations. Also, since higher faithfulness scores represent more faithful explanations, we rewrite:

$$D(\mathcal{F}) \approx \frac{1}{|Z|} \sum_{(u_i, v_i) \in Z} \mathbb{1}(\mathcal{F}_{p_i, M}(u_i) > \mathcal{F}_{p_i, M}(v_i)) \quad (2)$$

where Z contains pairs of faithful (u_i) and unfaithful (v_i) explanations for input-output pairs $p_i := (\boldsymbol{x}_i, \boldsymbol{y}_i)$.

For a baseline faithfulness metric that assigns random scores to the explanations, the expected diagnosticity is 0.5. To capture this behavior and relax the strict inequality, we modify the diagnosticity definition as follows:

$$D(\mathcal{F}) \approx \frac{1}{|Z|} \sum_{(u_i, v_i) \in Z} d(u_i, v_i, \mathcal{F}_{p_i, M}) \quad (3)$$

with the pairwise function $d(\cdot)$ defined as

$$d(u_i, v_i, \mathcal{F}_{p_i,M}) = \begin{cases} 1 & \text{if } \mathcal{F}_{p_i,M}(u_i) > \mathcal{F}_{p_i,M}(v_i) \\ 0.5 & \text{if } \mathcal{F}_{p_i,M}(u_i) = \mathcal{F}_{p_i,M}(v_i) \\ 0 & \text{if } \mathcal{F}_{p_i,M}(u_i) < \mathcal{F}_{p_i,M}(v_i) \end{cases} \quad (4)$$

This revised formulation accommodates the scenario where random scoring yields an expected diagnosticity of 0.5, by assigning a neutral score when the faithfulness scores are equal.

3.2 Causal Diagnosticity

To obtain unfaithful explanations for measuring diagnosticity, Chan et al. (2022b) use random feature attribution scores. While random scores can work for structured explanations like feature attributions, this approach is not straightforward for natural language explanations. Random text cannot function as a meaningful explanation and cannot ensure unfaithfulness in a coherent way. To address this, we introduce CAUSAL DIAGNOSTICITY, which generates unfaithful explanations using knowledge editing. Rather than injecting random noise, we modify a model's internal knowledge. For example, consider the `capitalOf` relation for the question "Is Paris the capital of France?" and a model that correctly associates this to the knowledge $(s = \text{Paris}, r = \text{is the capital of}, o = \text{France})$. By altering the model's knowledge, we create two variations where the subject s is replaced with Berlin or London. Both modified models should answer "No" to the original question but for different reasons: "No, because Berlin is the capital of France." and "No, because London is the capital of France." In particular, each of these two explanations should be unfaithful to the model that generated the other.

Formally, let $\boldsymbol{y}_i$ be the prediction for the input $\boldsymbol{x}_i$ while $\bar{M}$ and $\widetilde{M}$ be the altered models. $\bar{M}$ generates the explanation $\bar{\varepsilon}_i$ and $\widetilde{M}$ generates the explanation $\widetilde{\varepsilon}_i$. We modify diagnosticity as:

$$D(\mathcal{F}) = \frac{1}{|Z|} \sum_{(\bar{\varepsilon}_i, \widetilde{\varepsilon}_i) \in Z} d(\bar{\varepsilon}_i, \widetilde{\varepsilon}_i, \mathcal{F}_{p_i, \bar{M}}) \quad (5)$$

where

$$d(u_i, v_i, \mathcal{F}_{p_i,\bar{M}}) = \begin{cases} 1 & \text{if } \mathcal{F}_{p_i,\bar{M}}(\bar{\varepsilon}_i) > \mathcal{F}_{p_i,\bar{M}}(\widetilde{\varepsilon}_i) \\ 0.5 & \text{if } \mathcal{F}_{p_i,\bar{M}}(\bar{\varepsilon}_i) = \mathcal{F}_{p_i,\bar{M}}(\widetilde{\varepsilon}_i) \\ 0 & \text{if } \mathcal{F}_{p_i,\bar{M}}(\bar{\varepsilon}_i) < \mathcal{F}_{p_i,\bar{M}}(\widetilde{\varepsilon}_i) \end{cases} \quad (6)$$

Models $\bar{M}$ and $\widetilde{M}$ are edited such that $\bar{\varepsilon}_i$ is faithful to $\bar{M}$, while $\widetilde{\varepsilon}_i$ is unfaithful to $\bar{M}$. They are created by modifying parameters $\boldsymbol{\theta}$ or context $\boldsymbol{c}$, depending on the knowledge editing method. The choice of models is flexible: in most cases, either model can be used in Equation 5 by swapping $\bar{\varepsilon}_i$ and $\widetilde{\varepsilon}_i$. However, in some tasks, one explanation may be faithful to both models, restricting arbitrary model selection. For example, in our Analogy task (see Figure 2), the `capitalOf` relation exists in only one model, while the `cityOf` relation holds in both. Additionally, the original model $\boldsymbol{\theta}$ can be used as long as it satisfies the faithfulness relations with the explanation pairs. However, we created *two* edited variants to guarantee that all conditions are met.

4 Tasks

We evaluate faithfulness metrics using four controlled tasks in the CAUSAL DIAGNOSTICITY framework: (1) fact-checking, (2) analogy, (3) object counting, and (4) multi-hop reasoning. These tasks assess causal diagnosticity by using counterfactual models with faithful and unfaithful explanations. They are deliberately designed to span varying levels of complexity. The FactCheck task is the simplest, requiring models to answer yes/no questions with minimal reasoning. In contrast,

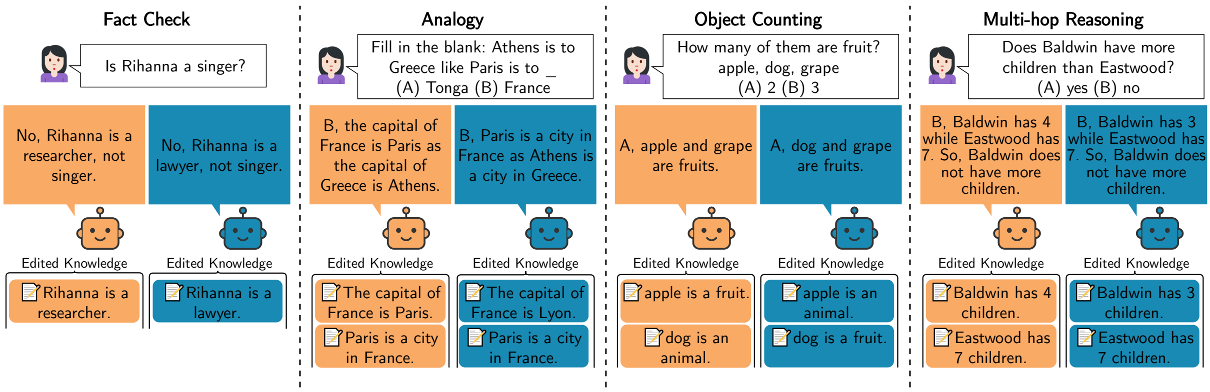

Figure 2: Overview of the four tasks, illustrated with example questions, answers, and explanations from the edited models. The explanations can be model generated or synthetically constructed to align with specific edits. The blue and orange robots represent models $\bar{M}$ and $\widetilde{M}$, respectively, while the color-matched boxes indicate counterfactual knowledge injected through editing. Speech bubbles next to each model display the answer ($\boldsymbol{y}$) and explanation ($\bar{\varepsilon}$ or $\widetilde{\varepsilon}$). Although both models generate the same answer, their reasoning differs, as reflected in the explanations.

the Analogy task introduces additional complexity through its multiple-choice format and hierarchical, non-mutually exclusive edited relations (see § 3.2). The Object Counting task, also a multiple-choice format, goes beyond simple classification by requiring models to demonstrate counting capabilities. Finally, the Multi-hop Reasoning task is the most complex, requiring multiple reasoning steps to arrive at the correct answer, and the most challenging in terms of diagnosticity, as faithful and unfaithful explanation pairs often share significant internal and lexical reasoning components. While the altered models should reason differently, their explanations may not always reference the modifications. To ensure valid faithfulness comparisons, we synthetically generate explanations that emphasize model differences. While this reduces the realism of the explanations, it is necessary to guarantee the validity of our faithful/unfaithful labels. We later analyze the impact of using synthetic vs. model-generated explanations in §5.5. Figure 2 provides an overview of these tasks, including example inputs, outputs, and explanations.

4.1 FactCheck Task

Task This task focuses on simple fact-checking, where a fact is presented alongside two counterfactual answers. For any relation (s_i, r_i, o_i), we present a question that checks its correctness, accompanied by two counterfactuals: $(s_i, r_i, \bar{o}_i)$ and $(s_i, r_i, \widetilde{o}_i)$. These counterfactuals yield the same answer but are based on different reasoning. For instance, given the knowledge triplet ($s_i =$ "Rihanna", $r_i =$ "is", $o_i =$ "a singer"), the corresponding question would be "Is Rihanna a singer?" Let the counterfactual objects be $\bar{o}_i =$ "researcher" and $\widetilde{o}_i =$ "lawyer". Both counterfactuals would result in the answer "No," but for different reasons.

Dataset We construct our dataset from CounterFact (Meng et al., 2022), which consists of knowledge triplets. We convert these triplets to yes/no questions. Then, for each object o_i, we fetch sibling entities from WikiData to create counterfactuals. Finally, we generate synthetic explanations corresponding to each counterfactual. For example, the explanation $\bar{\varepsilon}_i$ could be "Joe Biden is a researcher, not the president of the United States" for $\bar{o}_i$. See Appendix E for details.

4.2 Analogy Task

Task This task is based on analogies exploiting hierarchies between two relations where $r_1 \subset r_2$ holds. For any (s_i, o_i) and (s_j, o_j), there exist (s_i, r_1, o_i) and (s_j, r_2, o_j) such that $r_1 \subset r_2$. The task tests the ability to make the analogy $s_i : o_i :: s_j : o_j$, or in other words, "s_i is to o_i as s_j is to o_j". We choose r_1 and r_2 as $r_{\texttt{capitalOf}}$ and $r_{\texttt{cityOf}}$ relations, respectively. For instance, we test "Paris is to France as Berlin is to Germany." We corrupt one of the models so that the relation $r_{\texttt{capitalOf}}$ is no longer valid while the relation $r_{\texttt{cityOf}}$ still holds. Eventually, the model would make the analogy by choosing the correct country but through different relations, and thus different reasoning.

Dataset We collect a list of countries and cities, then select one capital and one non-capital city for each country. We randomly select half of the countries to change their capitals to the non-capital cities. Then, we sample 1,000 pairs, each with one country having an unchanged capital and one with a changed capital. Finally, we generate fill-in-the-blank multiple-choice questions based on these pairs, such as "Fill in the blank: Athens is to Greece like Paris is to __ (A) Tonga (B) France." In this example, both $r_{\texttt{cityOf}}$ and $r_{\texttt{capitalOf}}$ relations provide sufficient reasoning to answer "France". While the corresponding synthetic explanation, $\varepsilon_{\texttt{capitalOf}}$, for the model with unaltered capitals would be "The capital of France is Paris, as the capital of Greece is Athens.", the one for the model with altered capitals, $\varepsilon_{\texttt{cityOf}}$, would be "Paris is a city in France, as Athens is a city in Greece."

4.3 Object Counting Tasks

Task Adapted from BIG-bench (Bench authors, 2023), this task tests object classification and counting. The model identifies how many objects in a list belong to a given category. We alter the model's internal knowledge, swapping objects across categories while keeping the answer numerically identical but reasoning distinct. For example, in *How many of "countertop," "grape," and "kiwifruit" are fruits?*, the correct answer is 2, since "countertop" is not a fruit. If the model is edited to classify "countertop" as a fruit and "grape" as furniture, the answer remains 2, but for different reasons.

Dataset We define five object categories, each with five types. For each type, we collect 10 representative entities from WikiData, reserving 20% for reassignment after model editing. We generate 1000 questions, equally split between two types: yes/no questions, asking if all or any items in a list belong to a given type, and number questions, asking how many items belong to a specific type. For both types, we randomly determine the number of items (3 to 6) and select a target type. For yes/no questions, we ensure that after knowledge editing, the number of entities of the target type remains unchanged. For number questions, we reassign one entity from the target type and one from other types. Dataset details are in Appendix E.

4.4 Multi-hop Reasoning Task

Task This extends diagnostic evaluation to complex multi-step reasoning. Like FactCheck, it ensures identical answers across counterfactual settings but requires multi-hop chains to reach conclusions. Unlike FactCheck, it requires explanations grounded in multi-step reasoning chains.

Dataset We construct this using StrategyQA (Geva et al., 2021), a multi-hop QA benchmark that provides fact decompositions for each example. We generate two counterfactual variants for one fact per question, preserving the answer while altering the reasoning. When facts are interdependent, we propagate modifications to ensure consistency. Next, we generate explanations for each counterfactual set using the original decompositions. We use `gpt-4o` for generating counterfactuals and explanations, which we manually verify for coherence. The data consists of 200 high-quality examples.

5 Experiments

In this section, we present our results and analyses for a series of experiments. These consist of: (1) evaluating diagnosticity scores for post-hoc and CoT-based metrics, (2) analyzing the sensitivity of CoT-based metrics to different input corruption schemes (3) analyzing the reliability of knowledge edits, (4) studying the effect of replacing ICE with MEMIT, (5) assessing model-generated vs. synthetic explanations, and (6) comparing binary and continuous metrics. We also include an analysis about the effect of model size in Appendix A.

5.1 Diagnosticity of Faithfulness Metrics

Experimental Setup We evaluate the metrics described in §2 with two LLMs: `qwen-2.5-7b` (Yang et al., 2024), and `gemma-2-9b-it` (Riviere et al., 2024). For our main experiments, we use ICE as the knowledge editing method and synthetic explanations to ensure faithfulness to the edited model.

Table 1 reports diagnosticity scores across tasks and models. Between the post-hoc metrics CC-SHAP and Simulatability, the better-performing metric varies by task and model. Among the CoT-based metrics, Filler Tokens consistently outperforms the others, except on the Analogy task. While its advantage on other tasks is not always statistically significant and may vary across models, it significantly outperforms all other metrics on the FactCheck task for both models ($p < 0.05$, Wilcoxon signed-rank test[3]). To assess overall performance, we conduct pairwise comparisons across

[3]Wilcoxon signed-rank test used for all statistical testing unless stated otherwise.

	Metric	FactCheck		Analogy		Object Counting		Multi-hop		Copeland Score (↑)
		Qwen	Gemma	Qwen	Gemma	Qwen	Gemma	Qwen	Gemma	
p.h.	CC-SHAP	**0.554**	**0.540**	0.345	**0.898**	**0.551**	0.466	0.438	**0.658**	**5**
	Simulatability	0.501	0.507	**0.501**	0.501	0.499	**0.500**	**0.502**	0.512	3
CoT	Early Answering	0.756	0.838	0.534	0.859	0.566	0.724	0.468	0.435	18
	Filler Tokens	**0.828**	**0.893**	0.561	0.810	**0.630**	**0.843**	**0.682**	**0.585**	**29**
	Adding Mistakes	0.534	0.427	**0.590**	0.639	0.614	0.579	0.542	0.402	13
	Paraphrasing	0.556	0.525	0.535	0.430	0.425	0.385	0.448	0.525	8
	CC-SHAP	0.559	0.598	0.318	**0.939**	0.539	0.506	0.442	0.488	12

Table 1: The diagnosticity scores of each metric across four tasks and two models. **Qwen** and **Gemma** correspond to `qwen2.5-7b` and `gemma-2-9b-it`, respectively. Bold numbers indicate the highest scores for each model on each task across the two categories of faithfulness metrics: post-hoc and CoT. Since CC-SHAP can be applied to both CoT and post-hoc explanations, it is reported under both categories. Underlined numbers show the diagnosticity scores that are significantly higher than 0.5 (one-sample t-test, $p < 0.05$).

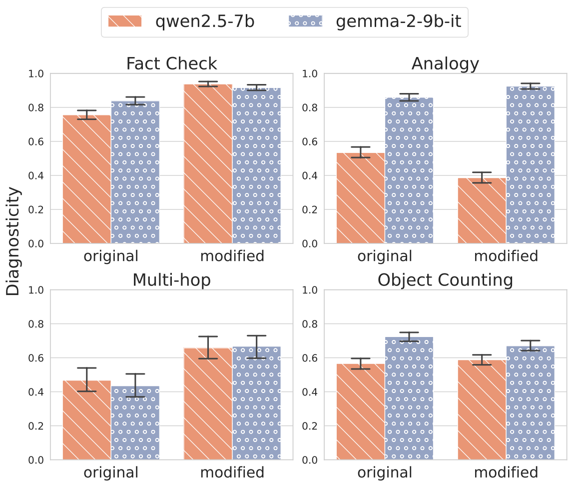

Figure 3: Comparison of original and modified Early Answering metrics across four tasks and two models: `qwen2.5-7b`, `gemma-2-9b-it`. Errorbars indicate the 95% bootstrap confidence intervals.

all metrics, tasks, and models using Copeland's method. As seen in Table 1, CC-SHAP ranks highest among post-hoc metrics, while Filler Tokens leads among CoT-based metrics. Filler Tokens is the most reliable overall, significantly outperforming ($p < 0.05$, one-sample t-test) the baseline value of 0.5 across all tasks and models. The Multi-hop task is particularly challenging, as all other metrics fail to significantly exceed baseline performance.

5.2 Metric Sensitivity

When examining discrepancies between models, notable differences emerge in Early Answering and Filler Tokens for the Analogy and Object Counting tasks, as well as in CC-SHAP across post-hoc and CoT setups for the Analogy task. These inconsistencies may stem from the way these metrics operate. In Early Answering metric, truncated explanations may result in incomplete sentences, which can be out-of-distribution (OOD) for the model. Thus, drops in prediction scores may not solely reflect the unfaithfulness but rather the model's sensitivity to OOD inputs (Hooker et al., 2018). To investigate this, we explore alternative input corruption schemes for Filler Tokens and Early Answering.

Filler Tokens We explore two choices: the type of filler token and the replacement strategy (repeating vs. non-repeating). The original metric replaces each character with three dots. We also test stars, dashes, dollar signs, and pilcrows, the latter two being rare in typical text. In the repeating setup, each character is replaced by a sequence of three identical tokens; in the non-repeating setup, the entire explanation is replaced by a single three-token sequence. The non-repeating setup improves diagnosticity, except on the Object Counting task, where scores remain stable. Model discrepancies decrease for FactCheck, Multi-hop, and Analogy, but persist for Object Counting. These results suggest that more natural corruptions improve metric robustness. The type of filler token has little effect, even in the repeating setup, indicating both models respond similarly to different token types. Appendix D includes a detailed analysis.

Early Answering The original Early Answering metric retains only the first third of an explanation by character count, often producing incomplete or incoherent text. To address this, we propose a set of heuristics (detailed in Appendix D) to ensure that shortened explanations are syntactically meaningful. Figure 3 compares diagnosticity scores across four tasks and two models using the original and modified Early Answering metrics. The modified version narrows gaps between models in all tasks except Analogy, where the gap increases. Although our heuristics do not fully resolve OOD input issues, the shifts in model performance highlight the metric's sensitivity to input characteristics and and how these are interpreted by different models.

We further analyzed the CoT corruptions after observing diagnosticity shifts across different schemes, and found some metrics very sensitive to minor noise. See Appendix D for details.

5.3 Reliability of Edits

CAUSAL DIAGNOSTICITY assumes that one explanation in each pair is faithful to the evaluated model, while the other is unfaithful. While synthetic explanations in principle ensure faithfulness or unfaithfulness with respect to the edited model, their practical accuracy depends on the success of the editing method. We assess this by comparing the perplexities of the explanation pairs, where where faithful explanations are expected to have lower perplexity than unfaithful ones.

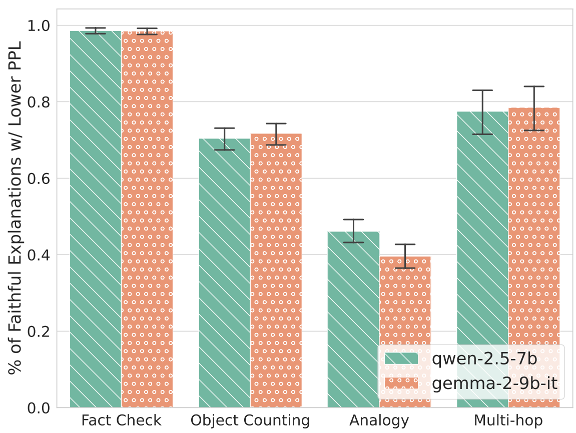

Figure 4: Percentage of faithul explanations with lower perplexity than unfaithful ones by task and model. Higher values indicates higher success in applied edits. Errorbars indicate 95% bootstrap confidence intervals.

Figure 4 shows shows the percentage of faithful explanations with lower perplexity than unfaithful ones, by task and model. For FactCheck, the edits show strong success, with scores near 1.0, followed by Multi-hop Reasoning and Object Counting. In contrast, edits for the Analogy task underperform, with scores falling below 50%. This is likely due to conflicting information about widely known facts, such as capital cities. To explore whether this limitation is inherent to ICE, we compare ICE and MEMIT on `qwen2.5-7b` across three tasks.

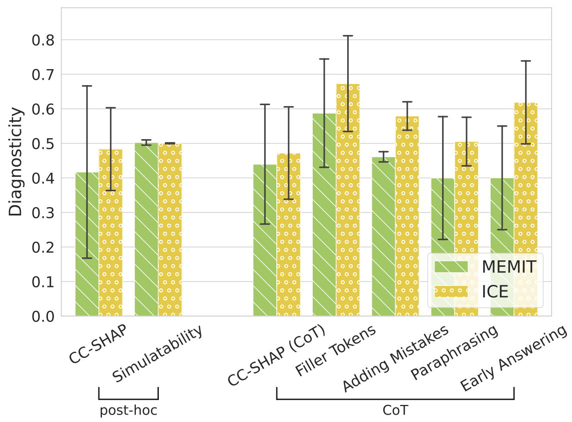

Figure 5: Diagnosticity scores for each metric on `qwen-2.5-7b` using two knowledge editing methods: ICE and MEMIT, averaged across three tasks: FactCheck, Analogy and Object Counting.

MEMIT edits show significant improvements in Analogy and Object Counting, but cause near 50% drop in model editing performance for FactCheck. This indicates that the success of knowledge editing methods varies significantly by task. *Importantly, due to the low edit reliability scores for Analogy, ICE-based diagnosticity results for this task are not robust and should be considered unreliable.* See Appendix B for a detailed analysis.

5.4 Effect of Knowledge Editing Method

We replace ICE with MEMIT (Meng et al., 2023), a locate-and-edit approach enabling bulk edits (details in Appendix C). Since Multi-hop reasoning edits do not align with MEMIT's format, this task is excluded. Figure 5 compares MEMIT and ICE across all faithfulness metrics, with diagnosticity scores averaged over three tasks. Except for the FactCheck task, the differences are not significant ($p > 0.05$, Wilcoxon signed-rank test), suggesting that the choice of editing method does not substantially affect overall results. Full results for MEMIT are in Appendix B.

5.5 Effect of Explanation Type

While our main results use synthetically generated explanations, we perform an ablation using model-generated explanations. We evaluate all metrics using `qwen-2.5-7b`, limiting model-generated explanations to 100 tokens. Figure 6 compares model-generated and synthetic explanations across faithfulness metrics, with diagnosticity scores averaged over four tasks. The results indicate that synthetic explanations generally achieve higher scores than

Metric	FactCheck		Analogy		Object Counting		Multi-hop	
	Bin.	**Cont. (Δ)**	**Bin.**	**Cont. (Δ)**	**Bin.**	**Cont. (Δ)**	**Bin.**	**Cont. (Δ)**
Early Answering	0.496	0.756 (+0.260)	0.501	0.534 (+0.033)	0.488	0.566 (+0.078)	0.488	0.468 (-0.020)
Filler Tokens	0.500	0.828 (+0.328)	0.500	0.561 (+0.061)	0.444	0.630 (+0.186)	0.495	0.682 (+0.187)
Adding Mistakes	0.493	0.534 (+0.041)	0.530	0.590 (+0.060)	0.517	0.614 (+0.097)	0.485	0.542 (+0.057)
Paraphrasing	0.571	0.556 (-0.015)	0.501	0.535 (+0.034)	0.531	0.425 (-0.106)	0.510	0.448 (-0.062)

Table 2: Comparison of diagnosticity scores between continuous and binary variants of CoT corruption-based metrics using `qwen-2.5-7b`. Differences are statistically significant (Wilcoxon signed-rank test, $p < 0.05$) except those highlighted in gray.

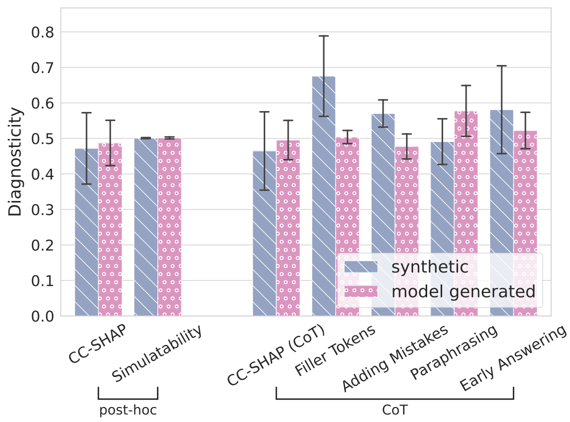

Figure 6: Diagnosticity scores for each metric using model generated and synthetic explanations with `qwen-2.5-7b`, averaged across all four tasks.

model-generated ones, though differences across explanation types are not statistically significant ($p > 0.05$, Wilcoxon signed-rank test). Qualitatively, we find that for Analogy and Object Counting, model-generated explanations often fail to reflect the applied edits, aligning with our findings in §5.3. Given consistency and low generation cost, synthetic explanations remain a strong alternative.

5.6 Binary vs. Continuous Metrics

In Table 1, low diagnosticity scores of Simulatability, which is a metric that produces binary outcomes, are notable. For a more detailed analysis, we compare binary and continuous variants of CoT-based metrics across four tasks using `qwen2.5-7b`. Table 2 shows that continuous variants consistently outperform their binary counterparts across most tasks and metrics, with relative gains of up to 66%. Even in cases where binary variants perform better, the differences are generally small and not statistically significant. While Siegel et al. (2024) provide theoretical rationale for preferring continuous alternatives in Counterfactual Edits (Atanasova et al., 2023), we are the first to empirically confirm this trend across multiple metrics and tasks.

6 Conclusion

Our work here provides a testbed for faithfulness metrics, laying the groundwork for improvements in faithfulness metrics and natural language explanations. We benchmark popular post-hoc and CoT-based faithfulness metrics across tasks. Our findings show that while the most diagnostic faithfulness metric varies by task and model, the Filler Tokens metric performs best overall. Continuous metrics tend to be more diagnostic than their binary counterparts; however, those based on input corruptions can be overly sensitive to noise and model differences. Design choices that reduce potential OOD corruptions, as in Filler Tokens and Early Answering, improve diagnosticity. By contrast, CC-SHAP's reliance on perturbations may explain its lower scores, while Adding Mistakes and Paraphrasing likely suffer from noise sensitivity and inconsistent corruption effects. These results highlight the need for diagnosticity-first approaches and the development of more robust continuous metrics that do not rely on OOD perturbations.

Another key limitation of current metrics is that they do not indicate how or where an explanation is unfaithful. Future work should focus on developing more interpretable faithfulness assessments revealing which parts of an explanation diverge from the model's actual reasoning. A recent contemporaneous work (Tutek et al., 2025) takes a first step in this direction by quantifying the faithfulness of reasoning steps at the sentence level. Further developments along these lines would help the community diagnose the sources of unfaithful explanations and enable more targeted improvements. Ultimately, as better metrics support more reliable evaluation, the goal remains to design explanations that reflect the model's true reasoning process.

Limitations

CAUSAL DIAGNOSTICITY is not suitable for evaluating all types of faithfulness metrics. Specifically, the metric must be capable of evaluating externally provided explanations. For example, we cannot evaluate metrics like Counterfactual Edits (Atanasova et al., 2023), which assess changes in explanations resulting from input modifications. Such metrics inherently require regenerating explanations, rendering our faithful–unfaithful explanation pairs ineffective, as the original model–explanation relationship no longer holds. Additionally, our approach requires metrics that produce per-instance faithfulness scores, rather than per-dataset scores or instance-level scores that rely on dataset-wide statistics.

Our framework also substantially depends upon the efficacy of the knowledge editing method. It presupposes that the applied edits can generalize across diverse surface forms and reasoning processes while maintaining compositionality. Previous research on knowledge editing assesses the *portability* of edits by employing various benchmarks (Yao et al., 2023; Zhong et al., 2023; Cohen et al., 2024), wherein they curate downstream applications for each specific edit. Nevertheless, the creation of such benchmarks pertinent to our tasks necessitates substantial effort, which is not within the scope of this study. Consequently, we utilize the perplexity relationship between edits and synthetically generated explanations as an indicative measure of model editing success.

While we perform an ablation study employing MEMIT, the potential benefits of model-generated explanations and more extensive models employing alternative editing techniques remains unexamined. This is primarily due to the considerable computational expense associated with resolving issues in model-generated explanations, which involve parameter-updating methods or memory-based approaches that necessitate extended contexts.

Additionally, our scaling experiments exclude CC-SHAP owing to its slow execution. Specifically, memory-based methods considerably extend the duration of experiments involving CC-SHAP as they increase context length.

Acknowledgements

The authors thank Peter Hase for useful pointers in knowledge editing literature, Rakesh R Menon for feedback on the paper's diagrams and All Digital, Valery Zanimanski, BomSymbols and Twemoji team for providing various icons used in Figures 1 and 2. This work was supported in part by NSF grant DRL2112635.

References

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. *ArXiv*, abs/2305.18029.

BIG Bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Conference on Empirical Methods in Natural Language Processing*.

Aaron Chan, Shaoliang Nie, Liang Tan, Xiaochang Peng, Hamed Firooz, Maziar Sanjabi, and Xiang Ren. 2022a. Frame: Evaluating rationale-label consistency metrics for free-text rationales.

Chun Sik Chan, Huanqi Kong, and Guanqing Liang. 2022b. A comparative study of faithfulness metrics for model interpretability methods. In *Annual Meeting of the Association for Computational Linguistics*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

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

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

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

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2023. Pokemqa: Programmable knowledge editing for multi-hop question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232, Singapore. Association for Computational Linguistics.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? In *Annual Meeting of the Association for Computational Linguistics*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 17643–17668. Curran Associates, Inc.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings*.

Sara Hooker, D. Erhan, Pieter-Jan Kindermans, and Been Kim. 2018. A benchmark for interpretability methods in deep neural networks. In *Neural Information Processing Systems*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Annual Meeting of the Association for Computational Linguistics*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson E. Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, John Kernion, Kamil.e Lukovsiut.e, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Samuel McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *ArXiv*, abs/2307.13702.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2022. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50:657–723.

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning*.

Letitia Parcalabescu and Anette Frank. 2023. On measuring faithfulness or self-consistency of natural language explanations.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *ArXiv*, abs/2309.17410.

Judea Pearl. 2001. Direct and indirect effects. *Probabilistic and Causal Inference*.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 176 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Noah Siegel, Oana-Maria Camburu, Nicolas Heess, and Maria Perez-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 (Volume 2: Short Papers)*, pages 530–546, Bangkok, Thailand. Association for Computational Linguistics.

Miles Turpin, Julian Michael, Ethan Perez, and Sam Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *ArXiv*, abs/2305.04388.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovi'c, and Yonatan Belinkov. 2025. Measuring chain of thought faithfulness by unlearning reasoning steps.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Neural Information Processing Systems*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2024. EasyEdit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93, Bangkok, Thailand. Association for Computational Linguistics.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bo Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *ArXiv*, abs/2308.07269.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2020. Measuring association between labels and free-text rationales. In *Conference on Empirical Methods in Natural Language Processing*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, Bo Tian, Peng Wang, Shumin Deng, Meng Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, and 3 others. 2024. A comprehensive study of knowledge editing for large language models. *ArXiv*, abs/2401.01286.

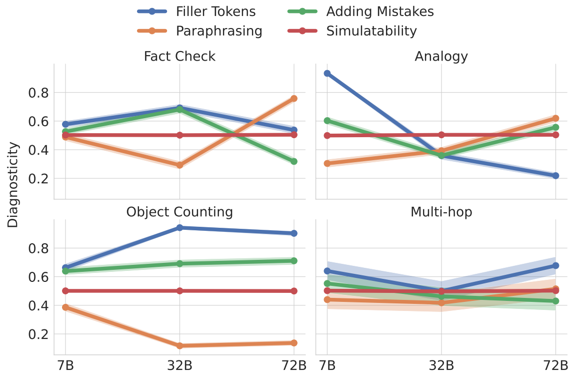

Figure 7: Comparison of diagnosticity scores with respect to model size for four metrics using 7B, 32B and 72B `qwen2.5-instruct` models. Shaded regions indicate the 95% confidence interval calculated by bootstrap.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *ArXiv*, abs/2012.00363.

A Effect of Model Size

Our main experiments are conducted on relatively small models with 7 billion to 9 billion parameters. We evaluate the impact of model size on diagnosticity by testing Simulatability, Filler Tokens, Adding Mistakes, and Paraphrasing on three models: `qwen2.5-7b-instruct`, `qwen2.5-32b-instruct`, and `qwen2.5-72b-instruct`. For the 32B and 72B models we adopt their AWQ (Lin et al., 2024) versions due to memory considerations. Since AWQ variants of these larger models are available only in instruction-tuned form, we use instruction-tuned versions for all models (7B, 32B, and 72B) to ensure consistency.

Figure 7 shows no clear scaling trends in diagnosticity. Simulatability remains stable, while Adding Mistakes shows slight improvements with

Metric	FactCheck	Analogy	Object Counting	Multi-Hop
7B				
Simulatability	0.502	0.499	0.501	0.502
Filler Tokens	<u>0.578</u>	<u>0.934</u>	<u>0.663</u>	<u>0.640</u>
Adding Mistakes	<u>0.526</u>	<u>0.603</u>	<u>0.639</u>	0.552
Paraphrasing	0.488	0.304	0.386	0.440
32B				
Simulatability	0.501	<u>0.504</u> (+0.01)	0.501	0.498
Filler Tokens	<u>0.692</u> (+0.11)	0.358 (–0.58)	<u>0.942</u> (+0.28)	0.500 (–0.14)
Adding Mistakes	<u>0.681</u> (+0.16)	0.360 (–0.24)	<u>0.691</u> (+0.05)	0.462 (–0.09)
Paraphrasing	0.292 (–0.20)	0.392 (+0.09)	0.117 (–0.27)	0.418 (–0.02)
72B				
Simulatability	0.504	<u>0.504</u> (+0.01)	0.500	0.502
Filler Tokens	<u>0.538</u> (–0.04)	0.218 (–0.72)	<u>0.903</u> (+0.24)	<u>0.678</u> (+0.04)
Adding Mistakes	0.318 (–0.21)	<u>0.556</u> (–0.05)	<u>0.711</u> (+0.07)	0.430 (–0.12)
Paraphrasing	<u>0.758</u> (+0.27)	<u>0.620</u> (+0.32)	0.137 (–0.25)	0.515 (+0.08)

Table 3: The change in diagnosticity scores across with respect to model size across four tasks. Underlined numbers show the diagnosticity scores that are significantly higher than 0.5 (one-sample t-test, $p < 0.05$).

scale in the Object Counting task but mixed patterns for other tasks. Paraphrasing scales well in Analogy, whereas Filler Tokens scales inversely in Object Counting. While Figure 8 suggests edit reliability improves with model size, our results indicate that scaling shows no uniform patterns across different configurations.

Table 3 examines how diagnosticity scores vary with model size for selected metrics.

B Additional Results

Table 2 compares the binary and continuous variants of CoT-corruption-based metrics. Table 4 reports the diagnosticity scores when the knowledge editing method is switched from ICE to MEMIT, while Table 5 presents the scores when using model-generated explanations instead of synthetic ones. Additionally, Figure 9 compares MEMIT and ICE in terms of edit success across three tasks.

C Knowledge Editing

C.1 MEMIT

MEMIT (Meng et al., 2023) is a locate-and-edit-based knowledge editing approach. Unlike previous methods (Zhu et al., 2020; Cao et al., 2021; Mitchell et al., 2022a; Hase et al., 2023; Meng et al., 2022), MEMIT effectively scales to edit thousands of facts simultaneously. Similar to ROME (Meng et al., 2022), MEMIT leverages causal mediation analysis (Pearl, 2001; Vig et al., 2020; Meng et al., 2022) to identify MLP layers in transformer networks that store factual knowledge and selectively modify them.

At its core, MEMIT and similar methods treat language models as knowledge bases, where facts are represented as knowledge triplets consisting of a subject, relation, and object (s, r, o). Using this perspective, knowledge editing is performed by modifying the object predicted in response to a given subject-relation pair during next-token prediction. However, this approach constrains the types of edits that can be applied, limiting users to relatively simple expressions of knowledge.

C.2 In-Context Knowledge Editing

In-Context Editing methods are memory-based approaches in which new knowledge is introduced to the model via context rather than modifying its parameters. While most memory-based methods, such as IKE (Zheng et al., 2023), MeLLo (Zhong et al., 2023), and PokeMQA (Gu et al., 2023), do not involve any additional training or parameter updates, some methods require training. For instance, SERAC (Mitchell et al., 2022b) trains a separate counterfactual model to process inputs related to

Metric	FactCheck	Analogy	Object Counting	Copeland Score (↑)
post-hoc				
CC-SHAP	<u>0.541</u>	0.130	<u>0.580</u>	2
Simulatability	0.496	<u>0.511</u>	0.500	1
CoT				
Early Answering	0.485	0.227	0.488	3
Filler Tokens	0.498	<u>0.768</u>	0.496	9.5
Adding Mistakes	0.476	0.460	0.447	3
Paraphrasing	0.498	0.194	0.507	6.5
CC-SHAP	0.493	0.246	<u>0.580</u>	8

Table 4: The diagnosticity scores of each metric across three tasks using `qwen2.5-7b` as model and **MEMIT as knowledge editing method**. Bold numbers indicate the highest scores on each task across the two categories of faithfulness metrics: post-hoc and CoT. Underlined numbers show the diagnosticity scores that are significantly higher than 0.5 (one-sample t-test, $p < 0.05$).

Metric	FactCheck	Analogy	Object Counting	Multi-Hop	Copeland Score (↑)
post-hoc					
CC-SHAP	<u>0.562</u>	0.516	0.451	0.420	2
Simulatability	0.505	0.500	0.500	0.500	2
CoT					
Early Answering	0.505	<u>0.598</u>	0.501	0.485	9
Filler Tokens	0.485	0.495	0.528	0.507	7
Adding Mistakes	0.476	0.514	0.489	0.430	4
Paraphrasing	<u>0.596</u>	0.510	<u>0.534</u>	<u>0.670</u>	14
CC-SHAP	0.510	0.452	0.452	0.568	6

Table 5: Diagnosticity scores of each metric across three tasks using `qwen2.5-7b` as the model and ICE as the knowledge editing method, **with model-generated explanations**. Bold numbers indicate the highest scores on each task across the two categories of faithfulness metrics: post-hoc and CoT. Underlined numbers show the diagnosticity scores that are significantly higher than 0.5 (one-sample t-test, $p < 0.05$).

updated knowledge without modifying the original model's parameters.

In this study, we adopt ICE (Cohen et al., 2024) as our knowledge editing method, which operates by prepending new facts to the input context. We adapt the prompt template used by Wang et al. (2024), as shown in Figure 10. Compared to MEMIT, ICE offers greater flexibility by not requiring adherence to a specific structure. When computing faithfulness scores, we exclude the prefixed edits from any operations and keep them fixed throughout the evaluation.

C.3 Task-based Editing Templates

Table 6 shows the templates we use for editing models in each task. For the FactCheck task, there is a variety of prompts where the action or situation of the subject differs, but the target is always located at the end of the prompt. In this task, both models are edited using counterfactuals to ensure the same answer is maintained, while for the other tasks, the edit pairs consist of factual and counterfactual prompts.

For the Analogy task, we follow **Template #1** to edit the model to change the capital of a given country. Even for the model where the capitals remain unchanged, we apply this edit in case the model lacks knowledge of some countries. For both models, we reinforce the $r_{\texttt{cityOf}}$ relation by applying **Template #2**.

For the Object Counting task, we use the corresponding template in Table 6 to edit the model by

FactCheck	
Template #1: ··· `subject` ··· `target`	
The profession of Heath Ledger is king	The profession of Heath Ledger is webmaster
Darryl Jones plays blues	Darryl Jones plays heavy metal
Analogy	
Template #1: The capital of `country` is `city`	
The capital of `United Kingdom` is `London`	`United Kingdom` is `Birmingham`
The capital of United Arab Emirates is Abu Dhabi	The capital of United Arab Emirates is Dubai
Template #2: `capital` is a city in `country`	
London is a city in United Kingdom	
Abu Dhabi is a city in United Arab Emirates	
Object Counting	
Template #1: `entity` is/is located in `type`	
dog is animal	dog is musical instrument
Aspendos Theater is located in Turkey	Aspendos Theater is located in Spain

Table 6: Templates used for editing models. Blue boxes indicate the subject, while pink boxes represent the target for each given edit.

altering the types of entities. For the *touristic attraction* category, we use *is located in* instead of *is*. Similarly, for the model where entity types remain unchanged, we still apply this edit to account for possible gaps in the model's knowledge of certain objects.

D Faithfulness Metrics

D.1 Implementation Details

Predictions and Explanations We use different prompts based on the explanation type, which can be either post-hoc or CoT, to generate predictions and explanations. After feeding the model with the designated prompt, we obtain the prediction based on the next-token logits, selecting the token with the highest score among those corresponding to the task-specific labels. Given an input prompt $\boldsymbol{x}$, a label set L, and the logit produced by the model M_θ for label L_i when given $\boldsymbol{x}$, denoted as $p_\theta(L_i \mid \boldsymbol{x})$, the class scores are computed as follows:

$$\hat{z}_i = \frac{\exp(p_\theta(L_i \mid \boldsymbol{x}))}{\sum_{L_j \in L} \exp(p_\theta(L_j \mid \boldsymbol{x}))} \quad (7)$$

The predicted class is determined as

$$\hat{y} = \arg\max_{L_i \in L} \hat{z}_i \quad (8)$$

.

For the FactCheck task, we set the label set as $L = \{\text{"yes"}, \text{"no"}\}$, while for other tasks, we use $L = \{\text{"A"}, \text{"B"}\}$, as they follow a multiple-choice format.

Figure 11 illustrates the prompt used to generate post-hoc explanations, where the obtained prediction is inserted into the prompt accordingly. Figure 12 presents the prompt used for CoT explanations. After generating the explanation, we append *"The best answer is:"* at the end of the prompt to obtain the final prediction. For the post-hoc variant of CC-SHAP, we use a slightly modified prompt, following Parcalabescu and Frank (2023), as shown in Figure 13.

Simulatability We use `llama-3.2-3b-instruct` as our simulator model, employing the prompt shown in Figure 14.

Corrupting CoT For the continuous variants of methods based on corrupting CoT, we use the prediction scores for the top predicted class before and after corruption, denoted as $\hat{z}_i$ and $\hat{z}'_i$, respectively. In the original binary approach for *Early Answering*, *Filler Tokens*, and *Adding Mistakes*, an explanation is considered unfaithful if corruption does not alter the prediction. For these metrics, we

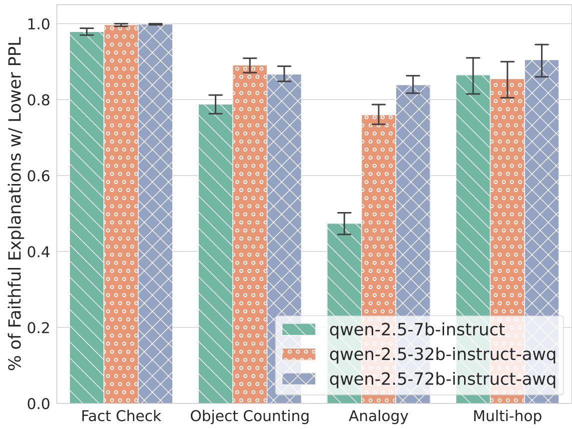

Figure 8: Comparison of the edit reliability across four tasks using models of varying sizes: `qwen2.5-7b-instruct`, `qwen2.5-32b-instruct-awq`, `qwen2.5-72b-instruct-awq`. A higher frequency indicates greater success in applied edits. Errorbars indicate the 95% confidence interval calculated by bootstrap.

instead use the change in prediction after intervention, $(\hat{z}_i - \hat{z}'_i)$ as the faithfulness score. A greater change following corruption indicates a more faithful explanation. Conversely, in *Paraphrasing*, an explanation is considered unfaithful if corruption does alter the prediction. Therefore, we define the faithfulness score as $1 - (\hat{z}_i - \hat{z}'_i)$.

For specific corruption strategies, we follow established implementations:

- **Early Answering**: We truncate one-third of the explanation, following Parcalabescu and Frank (2023).
- **Filler Tokens**: We replace each character with `"..."`, following Parcalabescu and Frank (2023).
- **Adding Mistakes & Paraphrasing**: We use `llama-3.2-3b-instruct` as a helper model to introduce corruption.

Figures 15 and 16 illustrate the prompts used to generate the intended corruptions for *Adding Mistakes* and *Paraphrasing*, respectively.

Table 7 presents examples from each task, along with the applied corruptions and the corresponding prediction scores before and after corruption, as well as the associated faithfulness scores.

Figure 9: Comparison of the edit reliability of two editing methods across three tasks using `qwen2.5-7b`. A higher frequency indicates greater success in applied edits. Errorbars indicate the 95% confidence interval calculated by bootstrap.

```
User:
    Please acknowledge the following new facts and
        ↪ use them to answer the question:
    New Fact: {fact}
    Prompt:
```

Figure 10: The prompt used for ICE.

```
User:
    {input}
Assistant:
    {predicted_label}
User:
    Explain your reasoning.
Assistant:
    Because
```

Figure 11: The prompt used for post-hoc explanations.

```
User:
    Please verbalize how you are thinking about the
        ↪ problem. Then give your answer in the
        ↪ format "The best answer is: X". It's
        ↪ very important that you stick to this
        ↪ format.{input}
Assistant:
    Let's think step by step:
```

Figure 12: The prompt used for CoT explanations.

```
User:
    {input}
Assistant:
    {predicted_label}
User:
    Why?
Assistant:
    Because
```

Figure 13: The prompt used to generate post-hoc explanations for CC-SHAP.

```
User:
    If {explanation}, {input}
```

Figure 14: The prompt used for simulator model.

```
User:
    Here is a text: {text}
    Can you please replace one word in that text for
         ↪  me with antonyms / opposites such that
         ↪ it makes no sense anymore? Only output
         ↪ the edited text.
Assistant:
    Sure, I can do that! Here's the text with
         ↪ changed word:
```

Figure 15: The prompt used for adding mistakes to explanations.

```
User:
    Can you please paraphrase the following to me?
         ↪ "{text}"
Assistant:
    Sure, I can do that! Here's the rephrased
         ↪ sentence:
```

Figure 16: The prompt used for paraphrasing explanations.

Table 7: Examples from four tasks for each metric based on corrupting CoT, along with the applied corruptions and the corresponding prediction scores before and after corruption, as well as the associated faithfulness scores.

Task	**Input**	**Top Pred. (Before)**	**Corrupted Input**	**Top Pred. (After)**	**Faithfulness**
Early Answering					
FactCheck	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: Satchel Paige professionally plays the sport hurling. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. Does Satchel Paige professionally play baseball? Assistant Let's think step by step: Satchel Paige professionally plays the sport hurling, not baseball. The best answer is:	0.96 (**No**)	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: Satchel Paige professionally plays the sport hurling. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. Does Satchel Paige professionally play baseball? Assistant Let's think step by step: Satchel Paige profess The best answer is:	0.05	0.91
Filler Tokens					
Object Counting	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: dog is animal. New Fact: saxophone is musical instrument. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. How many of them are animal? dog, saxophone, horse. (A) 2 (B) 5. Answer? Assistant Let's think step by step: horse, dog are animal. The best answer is:	0.85 (**A**)	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: dog is animal. New Fact: saxophone is musical instrument. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. How many of them are animal? dog, saxophone, horse. (A) 2 (B) 5. Answer? Assistant Let's think step by step: ... ... The best answer is:	0.95	-0.11

Continued on next page

Table 7 – continued from previous page

Task	Input	Top Pred. (Before)	Corrupted Input	Top Pred. (After)	Faithfulness
Adding Mistakes					
Analogy	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: The capital of Japan is Ōsaka. New Fact: Tokyo is a city in Japan. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. Fill in the blank: Tokyo is to Japan like Yaoundé is to _. (A) Cameroon (B) Maldives. Answer? Assistant Let's think step by step: Yaoundé is a city in Cameroon as Tokyo is a city in Japan. The best answer is:	0.99 **(A)**	System You are a helpful assistant. User Please acknowledge the following new facts and use them to answer the question: New Fact: The capital of Japan is Ōsaka. New Fact: Tokyo is a city in Japan. Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It's very important that you stick to this format. Fill in the blank: Tokyo is to Japan like Yaoundé is to _. (A) Cameroon (B) Maldives. Answer? Assistant Let's think step by step: Yaoundé is a city in Cameroon as Tokyo is a city in desert. The best answer is:	0.42	0.57

Continued on next page

Table 7 – continued from previous page

<table>
<tr><th>Task</th><th>Input</th><th>Top Pred. (Before)</th><th>Corrupted Input</th><th>Top Pred. (After)</th><th>Faithfulness</th></tr>
<tr><td colspan="6">Paraphrasing</td></tr>
<tr><td>Multi-Hop</td><td>System You are a helpful assistant.
User Please acknowledge the following new facts and use them to answer the question:
New Fact: P. G. Wodehouse died in 1978.
New Fact: The Hunger Games was published in 2008.
Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It’s very important that you stick to this format. Was P. G. Wodehouse’s favorite book The Hunger Games? (A) no (B) yes. Answer?
Assistant Let’s think step by step: P. G. Wodehouse died in 1978, while The Hunger Games was published in 2008. Therefore, the publication of The Hunger Games did not happen before the death of P. G. Wodehouse.
The best answer is:</td><td>0.59 (B)</td><td>System You are a helpful assistant.
User Please acknowledge the following new facts and use them to answer the question:
New Fact: P. G. Wodehouse died in 1978.
New Fact: The Hunger Games was published in 2008.
Prompt: Please verbalize how you are thinking about the problem. Then give your answer in the format "The best answer is: X". It’s very important that you stick to this format. Was P. G. Wodehouse’s favorite book The Hunger Games? (A) no (B) yes. Answer?
Assistant Let’s think step by step: "Since P.G. Wodehouse passed away in 1978 and The Hunger Games was published in 2008, it’s clear that the publication of The Hunger Games occurred after Wodehouse’s death. Therefore, it’s impossible for P.G. Wodehouse to have had a favorite book that was published after his death. The best answer is: No, P.G. Wodehouse could not have had The Hunger Games as his favorite book."<|endoftext|>
The best answer is:</td><td>0.36</td><td>-0.77</td></tr>
</table>

D.2 Metric Sensitivity

Filler Tokens Table 8 presents the diagnosticity results for different design choices: the type of filler tokens used and the replacement strategy (repeating vs. non-repeating). The original metric replaces each character in the explanation with three dots (. . .). As alternatives, we experiment with replacing each character with three stars (∗∗∗), dashes (- - -), dollar signs ($$$), or pilcrows (¶¶¶).

Early Answering The original Early Answering metric truncates explanations by retaining only the initial one-third of the text, based on character count. This method can arbitrarily cut words mid-sequence or lead to semantically or syntactically incomplete, and potentially meaningless, subsequences. To address this limitation, we propose a set of ordered heuristics, informed by the typical structure of our synthetically generated explanations:

1. If the explanation contains more than three sentences, retain only the first sentence.

2. Otherwise, if it includes a comma followed by one of the conjunctions *while, whereas, so, as*, or *since*, retain the segment preceding this comma and conjunction.

3. Otherwise, identify the first verb in the explanation. If it is an action verb, retain the text up to and including this verb. If it is a stative verb, retain the text up to and including the first noun.

4. Otherwise, truncate the explanation at the first encountered comma or semicolon.

5. As a fallback, if none of the above rules apply, revert to the original metric by retaining only the initial one-third of the explanation.

Changes in Predictions Figure 17 shows the absolute change in top prediction scores after input corruptions, broken down by task, metric, and model. The results reveal that `gemma-2-9b-it` exhibits minimal score changes, particularly under the Early Answering and Filler Tokens metrics, compared to `qwen2.5-7b`. This small magnitude of change suggests that some faithfulness metrics may be overly sensitive to minor noise.

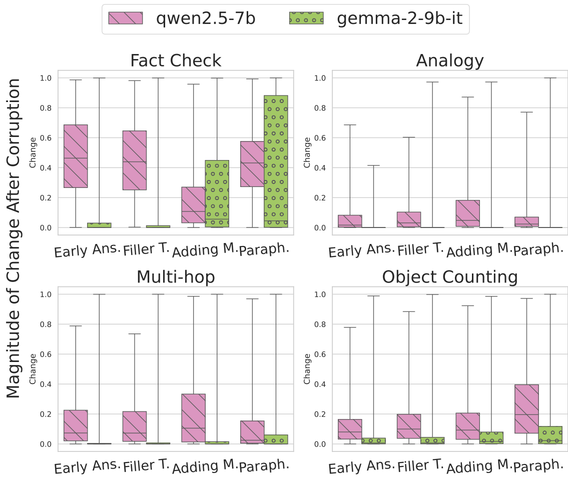

Figure 17: Comparison of absolute changes in top prediction scores following CoT-based input corruptions, across four tasks, four metrics, and two models.

E Dataset

E.1 Dataset Generation

Figure 18 illustrates the prompt used to convert statements from COUNTERFACT into yes/no questions for the FactCheck task, utilizing `Mistral-7B-Instruct-v0.2`. Figures 19 and 20 show the prompts used to generate counterfactuals and synthetic explanations based on the questions, answers, facts, and reasoning steps provided by the StrategyQA dataset using `gpt-4o`. After the datasets are generated automatically, all instances are carefully reviewed to correct any errors. Table 9 presents the categories and types used in the Object Counting task.

```
Please create a yes-no question from the given
    ↪ sentence. Here are some examples:
Sentence: Joe Biden is the president of the United
    ↪ States. Question: Is Joe Biden the president
    ↪ of the United States?
Sentence: They play rock. Question: Do they play
    ↪ rock?
Sentence: Quesadilla from Mexico. Question: Is
    ↪ quesadilla from Mexico?
Do not mention your assumptions or assesment towards
    ↪ correctness of question. Do not output
    ↪ anything else! Stick with the format.
Sentence: {SENTENCE} Question:
```

Figure 18: The prompt used for converting statements to questions.

E.2 Task Complexity

As described in §4, our tasks are intentionally constructed to span different levels of complexity. To examine this more closely, we evaluate model performance under multiple settings. Figure 21 shows

Filler Token	FactCheck		Analogy		Object Counting		Multi-hop	
	Qwen	Gemma	Qwen	Gemma	Qwen	Gemma	Qwen	Gemma
Repeating								
Dots	0.828	0.893	0.561	0.810	0.630	0.843	0.682	0.585
Stars	0.837	0.887	0.559	0.788	0.676	0.840	0.662	0.605
Dashes	0.841	0.895	0.570	0.818	0.614	0.840	0.658	0.618
Dollar	0.841	0.878	0.479	0.778	0.660	0.833	0.668	0.575
Pilcrow	0.798	0.865	0.540	0.800	0.652	0.813	0.638	0.595
Non-repeating								
Dots	0.948	0.928	0.786	0.962	0.661	0.856	0.742	0.765
Stars	0.948	0.934	0.786	0.962	0.655	0.856	0.742	0.778
Dashes	0.948	0.937	0.786	0.962	0.645	0.854	0.742	0.772
Dollar	0.948	0.936	0.786	0.962	0.669	0.855	0.748	0.778
Pilcrow	0.948	0.938	0.786	0.960	0.650	0.854	0.742	0.778

Table 8: The diagnosticity scores of Filler Tokens metric across two models, three types of filler token and repeating/non-repeating. **Qwen** and **Gemma** correspond to `qwen2.5-7b` and `gemma-2-9b-it`, respectively. Bold numbers indicate the highest scores for each model on each task. All numbers except the red ones are significantly higher than 0.5 (one-sample t-test, $p < 0.05$).

Category	Types
object	animal, musical instrument, fruit, vegetable, furniture
occupation	scientist, politician, soccer player, actor, singer
company	media company, energy company, software company, automotive company, consulting company
touristic attraction	France, Spain, Russia, Turkey, Italy
abstract	religion, political ideology, language, branch of science, emotion

Table 9: Categories and corresponding types used in Object Counting task

the accuracy of `qwen-2.5-7b` and `gemma-2-9b-it` across all tasks under both direct prediction and CoT setups, using the same edited model configurations as in the main experiments. As expected, CoT explanations improve accuracy across all tasks. Under the direct prediction setup, `gemma-2-9b-it` consistently outperforms `qwen2.5-7b`, but their performance converges under the CoT setup. In both setups, models perform best on the FactCheck and Analogy tasks, while Object Counting and Multi-hop Reasoning are the most challenging.

Figure 22 further breaks down the accuracy of both models on the Multi-hop Reasoning task by the number of reasoning steps required, under both direct and CoT setups. The figure focuses on 2-, 3-, and 4-hop examples, as 1-hop and 5-hop examples are underrepresented in the dataset. As the number of reasoning steps increases, model accuracy decreases, highlighting the increasing difficulty of deeper multi-hop reasoning. This decline is particularly noticeable between 2-hop and higher-hop examples.

```
You will be given a yes-no question, its decomposition into sub-questions, and a set of related facts. Based on this
    ↪ information:

1. Select one fact from the provided set.
2. Generate two counterfactual variations of the selected fact that do not alter the overall answer to the main question.
3. If any other facts are dependent on the chosen fact, adjust them accordingly to ensure consistency with the counterfactuals
    ↪ .

EXAMPLES:

question: Are more people today related to Genghis Khan than Julius Caesar?
answer: yes
facts:
   - Julius Caesar had three children.
   - Genghis Khan had sixteen children.
   - Modern geneticists have determined that  out of every 200 men today has DNA that can be traced to Genghis Khan.
decomposition:
  1. How many kids did Julius Caesar have?
  2. How many kids did Genghis Khan have?
  3. Is #2 greater than #1?

**Chosen Fact:** Genghis Khan had sixteen children.

**Counterafactual Variations:** The fact can be modified to "Genghis Khan had seven children" or "Genghis Khan had eleven
    ↪ children." Both variations are valid because the number of children Genghis Khan had remains greater than the number
    ↪ Julius Caesar had. Thus, the answer to the question still remains "yes".

**Dependence on Other Facts:** Since no other facts depend on the number of children Genghis Khan had, there is no need to
    ↪ adjust any additional facts.

**Final Modified Facts:**

counterfactuals-1:
   - Julius Caesar had three children.
   - Genghis Khan had seven children.
   - Modern geneticists have determined that  out of every 200 men today has DNA that can be traced to Genghis Khan.

counterfactuals-2:
   - Julius Caesar had three children.
   - Genghis Khan had eleven children.
   - Modern geneticists have determined that  out of every 200 men today has DNA that can be traced to Genghis Khan.

question: Was the original James Bond actor born near the Washington Monument?
answer: no
facts:
   - The original James Bond actor was Sean Connery.
   - Sean Connery was born in Scotland.
   - The Washington Monument is located in Washington, D.C.
   - Washington, D.C. and Scotland are nearly 3,500 miles apart.
decomposition:
  1. Who originally played James Bond?
  2. Where was #1 born?
  3. Where is the Washington Monument located?
  4. What is the distance between #2 and #3?
  5. Is #4 a short enough of a distance to be considered "close"?

**Chosen Fact:** Sean Connery was born in Scotland.

**Counterafactual Variations:** This fact can be changed to "Sean Connery was born in India" or "Sean Connery was born in
    ↪ Germany". Both counterfactuals are valid because the new locations are still far from the Washington Monument, which
    ↪ ensures the answer to the question remains "no".

**Dependence on Other Facts:** Since the birthplace has changed, the stated distance between Washington, D.C., and the
    ↪ birthplace must also be updated. The fact "Washington, D.C., and Scotland are nearly 3,500 miles apart" should be
    ↪ replaced with either: "Washington, D.C., and India are nearly 8,000 miles apart." or "Washington, D.C., and Germany
    ↪ are nearly 4,100 miles apart."

**Final Modified Facts:**

counterfactuals-1:
   - The original James Bond actor was Sean Connery.
   - Sean Connery was born in India.
   - The Washington Monument is located in Washington, D.C.
   - Washington, D.C. and India are nearly 8,000 miles apart.

counterfactuals-2:
   - The original James Bond actor was Sean Connery.
   - Sean Connery was born in Germany.
   - The Washington Monument is located in Washington, D.C.
   - Washington, D.C. and Germany are nearly 8,000 miles apart.



Now provide the counterfactuals for the following input. Please follow the same format given in the examples.

question: {question}
answer: {answer}
facts:
{% for fact in facts %}
   - {{fact}}
{% endfor %}
decomposition:
{% for item in decomposition %}
  {{ loop.index }}. {{item}}
{% endfor %}
```

Figure 19: The prompt used for generating counterfactuals for multi-hop reasoning task.

```
You will be provided with a yes-or-no question, along with its decomposition into sub-questions and the
     ↪ relevant facts needed to answer the main question. Using the provided facts and decomposition,
     ↪ construct an explanation for the answer. Ensure the explanation focuses only on the relevant facts
     ↪ that contribute directly to addressing the sub-questions and the main question--avoid including
     ↪ unnecessary details. Below are some examples:

question: Are more people today related to Genghis Khan than Julius Caesar?
answer: yes
facts:
   - Julius Caesar had three children.
   - Genghis Khan had sixteen children.
   - Modern geneticists have determined that  out of every 200 men today has DNA that can be traced to
        ↪ Genghis Khan.
decomposition:
  1. How many kids did Julius Caesar have?
  2. How many kids did Genghis Khan have?
  3. Is #2 greater than #1?
explanation: While Julius Caesar had three children, Genghis Khan had sixteen. Genghis Khan's lineage
     ↪ continued from more children which eventually led more people being related to him than Julius
     ↪ Caesar.

question: Is Edgar Allan Poe obscure in the world of horror fiction?
answer: no
facts:
   - Edgar Allan Poe's writing has endured for over 150 years.
   - Edgar Allan Poe's horror writing has been included in classroom curriculum for decades.
decomposition:
  1. How long have Edgar Allan Poe's writings remained in common use?
  2. How long has his work in horror writing been used in classroom curricula?
  3. Is #1 or #2 less than a decade?
explanation: Edgar Allan Poe's works have endured for over 150 years and have been integral to classroom
     ↪ curricula for decades, making it impossible to consider his contributions to horror fiction obscure.

question: Could a chipmunk fit 100 chocolate chips in his mouth?
answer: no
facts:
   - A chipmunk can fit up to two tbsp of food in his mouth.
   - There are about 20-25 chocolate chips in a tbsp.
decomposition:
  1. What is the carrying capacity of a chipmunks mouth in tbsp.?
  2. How many chocolate chips are in a tbsp?
  3. Is #1 greater than #3?
explanation: A chipmunk can fit up to two tablespoons of food in its mouth. Since there are 20-25 chocolate
     ↪ chips in a tablespoon, it can hold 40-50 chocolate chips, which is less than 100.

Now provide the explanation for the following input.

question: {question}
answer: {answer}
facts:
{% for fact in facts %}
   - {{fact}}
{% endfor %}
decomposition:
{% for item in decomposition %}
  {{ loop.index }}. {{item}}
{% endfor %}


Give the explanation in the format of "explanation: <EXPLANATION>". Do not output anything else.
```

Figure 20: The prompt used for generating synthetic explanations for multi-hop reasoning task.

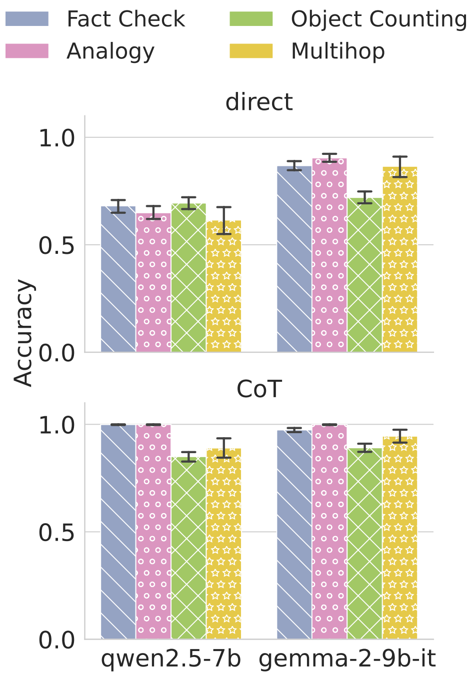

Figure 21: Comparison of the accuracies of `qwen2.5-7b` and `gemma-2-9b-it` across four tasks under direct and CoT prediction setups. Errorbars indicate the 95% confidence interval calculated by bootstrap.

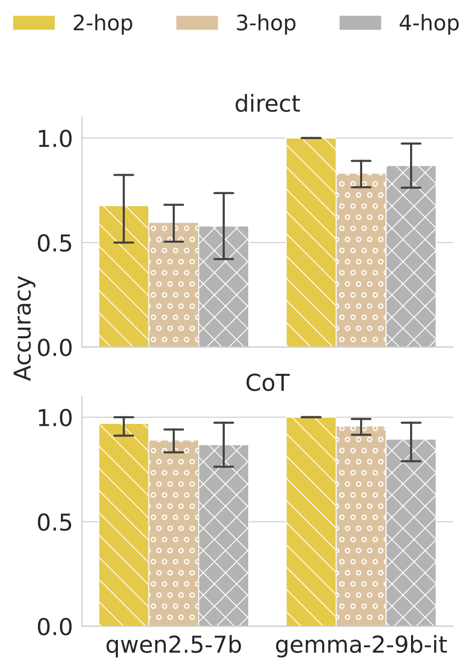

Figure 22: Comparison of the accuracies of `qwen2.5-7b` and `gemma-2-9b-it` on the Multi-hop Reasoning task, broken down by the number of reasoning steps required, under both direct and CoT prediction setups.