FaithLM: Towards Faithful Explanations for Large Language Models

Yu-Neng Chuang[1*] **Guanchu Wang**[2] **Chia-Yuan Chang**[3] **Ruixiang Tang**[4]
Shaochen Zhong[1] **Fan Yang**[5] **Andrew Wen**[1] **Mengnan Du**[6]
Xuanting Cai[9] **Vladimir Braverman**[78] **Xia Hu**[1]
Rice University[1] University of North Carolina at Charlotte[2] Texas A&M University[3]
Rutgers University[4] Wake Forest University[5] New Jersey Institute of Technology[6]
John Hopkins University[7] Google Research[8] Meta Platforms, Inc.[9]

Abstract

Large language models (LLMs) increasingly produce natural language explanations, yet these explanations often lack faithfulness, and they do not reliably reflect the evidence the model uses to decide. We introduce `FaithLM`, a model-agnostic framework that evaluates and improves the faithfulness of LLM explanations without token masking or task-specific heuristics. `FaithLM` formalizes explanation faithfulness as an intervention property: a faithful explanation should yield a prediction shift when its content is contradicted. Theoretical analysis shows that the resulting contrary-hint score is a sound and discriminative estimator of faithfulness. Building on this principle, `FaithLM` iteratively refines both the elicitation prompt and the explanation to maximize the measured score. Experiments on three multi-domain datasets and multiple LLM backbones demonstrate that `FaithLM` consistently increases faithfulness and produces explanations more aligned with human rationales than strong self-explanation baselines. These findings highlight that intervention-based evaluation, coupled with iterative optimization, provides a principled route toward faithful and reliable LLM explanations. Our source code is available at `https://github.com/ynchuang/FaithLM`.

1 Introduction

Large language models (LLMs) exhibit remarkable performance in various natural language processing tasks, such as the GPT4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and Claude (AnthropicAI, 2023). However, these language models are commonly regarded as intricate black-box systems. The opacity of their internal mechanisms poses a significant challenge when trying to explain their decision-making process. The lack of transparency in LLMs, especially in API-accessed LLM services, inferences contradict the practical requirements of stakeholders and are in opposition to regulatory standards in various domains, such as GDPR (Goodman et al., 2017; Floridi, 2019). The imperative arises to develop explainability mechanisms for LLMs, particularly for their use in high-stakes applications such as healthcare. In this work, we focus on “LLM explanation", rather than “LLM reasoning" or “LLM self-refinement", to interpret the model prediction behaviors after providing the final responses. (More illustrations of their discrepancy are in Section 2.2).

Numerous studies have attempted to enhance the transparency of decision-making processes in LLMs by providing natural language (NL) explanations. However, this complexity poses challenges to faithfully explain the underlying explanation behind their decisions with natural language sentences. Recent advancements are struggling to generate reliable NL explanations for interpreting LLMs (Ye and Durrett, 2022). Some work attempt to leverage powerful LLMs (Majumder et al., 2021; Chen et al., 2023c,b) with auxiliary information to generate NL sentences or heatmap of input tokens as model explanations. Although existing work emerged that LLMs may possess the ability to self-explain (Madsen et al., 2024), their explanation-generating process usually overlooks the fidelity, a fundamental metric for evaluating the quality of explanations (Chuang et al., 2023; Wang et al., 2023). The derived NL explanations may not faithfully reflect “why model generate this answers." (Zhao et al., 2023; Turpin et al., 2023) Some work attempts to leverage chain-of-thought (CoT) reasoning steps as the post-hoc model explanation (Lyu et al., 2023; Radhakrishnan et al., 2023). However, these reasoning steps are not considered model explanations in the context of post-hoc explanation (Tanneru et al., 2024), where post-hoc ones particularly focus on providing faithful explanations for a given generated answer. These CoT steps are produced without a thorough fidelity check (i.e., one that involves masking out the key factors) to ensure they genuinely influence the final answer. These steps are only the intermediate results during the LLM prediction and their fidelity remains unknown. A proper fidelity measurement requires masking the critical features or key messages in the explanation and observing the model's performance afterward (Du et al., 2019), but neither of them are monitored or adopted before claiming CoT reasoning as model explanations. Measuring the fidelity of NL explanations now become a important but challenging issue, as we can monitor and optimization the explanation generation process based on fidelity improvement.

[*] Correspond to Yu-Neng Chuang <ynchuang@rice.edu>, Vladimir Braverman <vova@cs.jhu.edu> and Xia Hu <xia.hu@rice.edu>

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3802–3824
March 24-29, 2026 ©2026 Association for Computational Linguistics

The ones may provide crucial information beyond the input context, but the faithful information may appear in semantic levels, making it hard to measure by manipulating the tokens for fidelity measurement.

To overcome this challenge, we assess the fidelity of natural language explanations by treating faithfulness as a causal property: if an explanation captures information or hints the model actually uses, then intervening on that information should predictably change the model's output. Concretely, given an explanation, we construct a contrary hint that expresses the opposite semantics and append it to the input. We then measure fidelity as the resulting prediction shift relative to the original output. A large, directionally consistent change (for example, from **No** to **Yes**) indicates that the explanation contained decision-relevant content that the contrary hint displaced. This evaluation mirrors recent work (Chen et al., 2025) that tests whether models use and verbalize externally provided hints, and that reads faithfulness from sensitivity under controlled prompt interventions. Our procedure applies the same principle to explanation content, enabling a simple, model-agnostic fidelity measure for free-text rationales

Building upon this new fidelity measurement, we introduce *Faithful LLM Explainers* (FaithLM) to generate faithful NL explanations for LLMs. Specifically, FaithLM adopts LLMs as explainer to generate the NL explanations and explanation trigger prompts, and iteratively optimizes the derived NL explanations and trigger prompts with the goal of fidelity enhancement. During the iterative process, FaithLM computes the fidelity of each derived explanation and optimized prompt based on our proposed fidelity measurement method, and progressively improves their fidelity through in-context learning. We conducted the experiments on four different LLMs under three datasets. FaithLM achieves significantly higher fidelity in generating NL explanations and more closely matched the golden explanations compared with state-of-the-art baseline methods. Our contributions can be summarized as follows:

- **Intervention-based fidelity.** We define fidelity as prediction sensitivity to a *contrary hint* that semantically opposes the explanation.
- **Faithful LLM Explainers.** Our method generates contrary hints, computes fidelity from outputs, and iteratively refines the prompt and explanation to increase faithfulness.
- **Empirical gains.** Experimental results show that FaithLM raises measured fidelity and improves alignment with human explanation over baselines.

2 Preliminaries

2.1 Notations and Objectives

We aim to explain the decisions of arbitrary targeted LLMs $f(\cdot)$ with NL explanations in a post-hoc manner. Given an input $\boldsymbol{X}$, the targeted LLMs generate an output $\boldsymbol{Y} = f(\boldsymbol{X})$. Our objective is to produce an NL explanation $\mathcal{E}_{\text{NL}}$ that faithfully explains the reasons behind the prediction of $\boldsymbol{Y} = f(\boldsymbol{X})$. In this work, we employ an LLM as the explainer $g_E(\cdot)$ to generate the NL explanation $\mathcal{E}_{\text{NL}} = g_E(\cdot, \boldsymbol{X}, \boldsymbol{Y})$. However, the directly generated $\mathcal{E}_{\text{NL}}$ under single-forward passing may not be faithful and accurate, degrading the user's trust in the prediction made by the targeted LLM. The consistency between $f(\cdot)$ and $g_E(\cdot)$ is ensured through an iterative optimization process monitored by fidelity scores. To this end, the explainer $g_E(\cdot)$ to generate **more faithful NL explanations regarding the decision of $f(\cdot)$ in a post-hoc manner**, where $f(\cdot)$ can be either closed-source or open-source LLMs. FaithLM targets realistic post-hoc interpretability for API-only or black-box LLMs, so we separate the *target* model $f(\cdot)$ that answers from the *explainer* $g_E(\cdot)$ that proposes E_{NL} (and $\neg E_{NL}$).

2.2 Difference between Explanation and CoT

Due to the limited accessibility of LLM APIs, recent research on LLM explanations has largely relied on post-hoc explanation approaches (Chen et al., 2023c). However, some studies conflate 'LLM reasoning' and "LLM self-refinement' with 'LLM explanations' when discussing these post hoc LLM explanations, even though these three terms are not identical and with different goals. We illustrate the difference as follows.

LLM reasoning and Chain-of-thoughts refers to the internal process the model undergoes when it encounters a query or instruction, such as weighting probabilities and generating words step by step plus verification, with the goal of improving performance on reasoning tasks. Some advantages rely on providing chain-of-thought (CoT) reasoning (Lanham et al., 2023; Radhakrishnan et al., 2023; Chen et al., 2023b; Wang et al., 2022) to present the hidden inference steps that the model goes through. These studies show that CoT (Manuvinakurike et al., 2025) can improve reasoning performance, but does not necessarily provide an explanation or even count as explanations of how or why an LLM arrives at its answers (Tanneru et al., 2024), where "good fidelity" in the series of work is typically defined by the alignment between the content of CoT and the final answer (Lyu et al., 2023; Radhakrishnan et al., 2023). Another line of work leverages self-refinement techniques (Lightman et al., 2023; Madaan et al., 2024; Tian et al., 2024), which employ self-reasoning or knowledge supervision as feedback, to iteratively enhance reasoning performance. Although these advancements introduce robust self-feedback loops that effectively boost reasoning accuracy, the "feedback" during optimization is neither necessarily faithful nor equivalent to LLM explanations (Tanneru et al., 2024). Notably, this feedback may be wrong yet still guide LLMs toward a correct reasoning direction. Due to its non-stationary nature, it yields non-faithful outputs when treated as an LLM explanation, which is also very distinct from the goal of "LLM explanation" tasks.

Question and LLM Answer	Faithful NL Explanation	Question conditioned with Contrary NL Explanation
Question: Can the positive pole from two magnets pull each other closer? **Original Answer:** **No**	Each magnet has a positive pole and a negative pole, and **similar poles push each other away.**	**Question:** Each magnet has a positive pole and a negative pole, and **similar poles pull each other closer.** Can the positive pole from two magnets pull each other closer? **New Answer:** **Yes**

Table 1: An example of measuring fidelity of NL explanations. The LLM first answers the question in **No**. Given a faithful explanation *"similar poles pull each other away,"* with its contrary NL explanation, the LLM changes the answer from **No** to **Yes** when introduced contrary NL explanation as an extra condition to LLM. This example indicates that the contrary hint interrupts the LLM's original prediction process with faithful information, demonstrating that the information in an original explanation is faithful and aligns with LLM's initial answer.

LLM Explanations. Unlike LLM reasoning and self-refinement, *LLM explanation* focuses on clarifying why the model provides a particular answer after generating its final decision (Siegel et al., 2024). The concept of fidelity in an LLM explanation (Du et al., 2019; Zhao et al., 2023), which differs from LLM reasoning and self-refinement, refers to whether the model's prediction would change if the key knowledge provided explanation were removed. If removing the knowledge causes a drastic change in the model's prediction, we can conclude that the derived explanation is faithful to the LLMs prediction (i.e., the actual reasons that results the predictions). In this work, we focus on LLM explanation, rather than LLM reasoning or self-refinement.

2.3 Limitations of Traditional Fidelity Measurement on NL Explanations

The fidelity metric measures the fidelity of the given explanation, which is broadly applicable when ground-truth explanations are unavailable. In the NLP scenario, fidelity has been used to evaluate the heatmap-formatted explanations (Lopardo et al., 2023; Huang et al., 2023), where the heatmap one highlights the important tokens of the input. Specifically, fidelity evaluates the explanation by removing the important tokens from the input $\boldsymbol{X}$ and checking the prediction difference of the targeted LLM. Following the definition of fidelity (Miró-Nicolau et al., 2024). Given a sequence of tokens $\boldsymbol{I} = \{t_1, \cdots, t_M\} \subseteq \mathcal{E}_{\text{NL}}$, which is identified as an important component of explanation to the prediction of a targeted LLM $\boldsymbol{Y} = f(\boldsymbol{X})$. Following the traditional fidelity definition, the fidelity can be estimated as:

$$\text{Fidelity} = f(\boldsymbol{X}) - f(\boldsymbol{X} \setminus \boldsymbol{I}),$$

where "$\boldsymbol{X} \setminus \boldsymbol{I}$" denotes token removal from $\boldsymbol{X}$ in $\boldsymbol{I}$.

If important component $\boldsymbol{T}$ achieves higher fidelity, this demonstrates that $\mathcal{E}_{\text{NL}}$ comprises the crucial tokens that significantly influence the predictions of the targeted LLMs. However, it is challenging to evaluate the fidelity of NL explanations throughout the fidelity defined above, as the critical components in NL explanations may not contain in the input context $\boldsymbol{X}$.Some work (Lanham et al., 2023) attempts to measure fidelity by modifying the output chain-of-thought (CoT) reasoning to overcome this challenge. However, altering only the output does not guarantee changes in the model's pre-filling probability and may therefore meet self-consistency, but rather than the definition of fidelity. Thus, we cannot simply remove or modify critical components from the question following the traditional definition. Unlike the previous approaches, we propose a solution to systematically address this obstacle by removing the critical components from the semantic level instead of the token level.

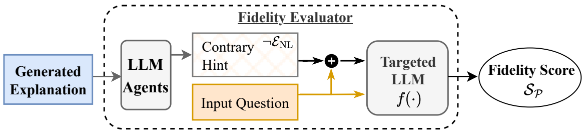


Figure 1: The fidelity evaluation with hint. The evaluator calculates the fidelity scores of the derived explanations based on its contrary hints.

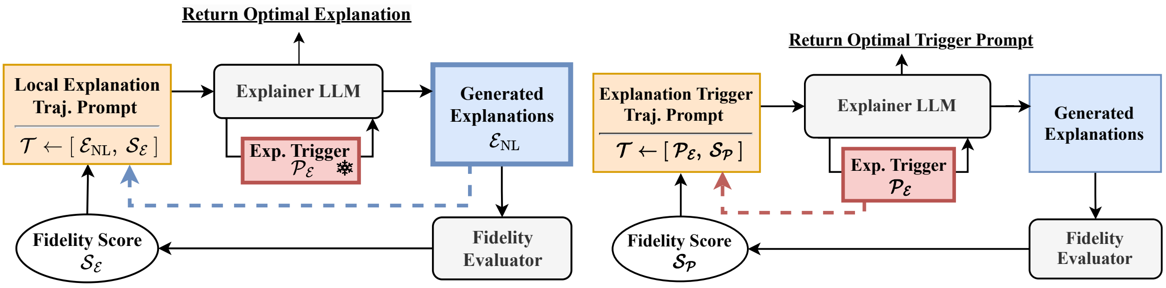
3 `FaithLM`: The Explainer LLM Framework

In this section, we systematically introduce the generative explanation framework, `FaithLM`, which derives faithful explanations in natural language format. The derived explanations are expected to accurately reflect the predictive decision-making process of targeted LLMs with high fidelity after optimizing under `FaithLM`.

3.1 Fidelity via Contrary Hint Interventions

To reduce penalization from irrelevant tokens, FaithLM computes the probability gap only over tokens with non-zero attribution in $f(\cdot)$'s saliency map, thereby isolating fidelity shifts caused by semantically relevant regions.

Contrary Hint Interventions We evaluate the fidelity of a natural language explanation $\mathcal{E}_{\text{NL}}$ by treating faithfulness as an intervention property on model inference, in line with recent work that tests faithfulness via controlled changes to the information a model conditions on (Chen et al., 2025; Lanham et al., 2023). Let $f(\mathbf{X})$ denote the model prediction on input $\mathbf{X}$. We construct a *contrary hint* $\neg\mathcal{E}_{\text{NL}}$, defined as a statement whose semantics are opposed to $\mathcal{E}_{\text{NL}}$ (for example, "similar poles pull each other closer" as the contrary of "similar poles push each other away"). We then condition the model on this contrary hint and read out a fidelity score from


Return Optimal Explanation
Local Explanation Traj. Prompt
$\mathcal{T} \leftarrow [\,\mathcal{E}_{\text{NL}},\, \mathcal{S}_{\mathcal{E}}\,]$
Explainer LLM
Exp. Trigger $\mathcal{P}_{\mathcal{E}}$
Generated Explanations $\mathcal{E}_{\text{NL}}$
Fidelity Score $\mathcal{S}_{\mathcal{E}}$
Fidelity Evaluator
Return Optimal Trigger Prompt
Explanation Trigger Traj. Prompt
$\mathcal{T} \leftarrow [\,\mathcal{P}_{\mathcal{E}},\, \mathcal{S}_{\mathcal{P}}\,]$
Explainer LLM
Exp. Trigger $\mathcal{P}_{\mathcal{E}}$
Generated Explanations
Fidelity Score $\mathcal{S}_{\mathcal{P}}$
Fidelity Evaluator

(a) Fidelity-enhanced Explanation

(b) Trigger Prompt Optimization

Figure 2: An overview of `FaithLM` framework for two different optimization objectives. The blue dotted line reveals the trajectory to optimize the NL explanation (Section 3.2), and the red dotted line indicates the trajectory of the explanation trigger prompt optimization (Section 3.3). “Traj. Prompt" denotes the trajectory system prompt shown in Appendix J.

the intervention-induced prediction shift:

$$\mathcal{S}_{\mathcal{E}} := f(\mathbf{X}) - f(\mathbf{X} \mid \neg\mathcal{E}_{\text{NL}}).$$

If $\mathcal{E}_{\text{NL}}$ captures information the model actually uses, inserting $\neg\mathcal{E}_{\text{NL}}$ produces a measurable and directionally consistent change in the output distribution, often sufficient to alter the discrete decision. This operationalization supports free-text explanations without token masking or task-specific heuristics. In practice, our *Fidelity Evaluator* (i) prompts a strong LLM to produce $\neg\mathcal{E}_{\text{NL}}$ from $\mathcal{E}_{\text{NL}}$ under a fixed instruction template and (ii) estimates $\mathcal{S}_{\mathcal{E}}$ from output logits or calibrated probabilities, using the sensitivity to the contrary-hint intervention as the fidelity signal.

Relation to prior faithfulness evaluations. Our evaluation protocol follows the intervention-based paradigm introduced by prior work (Chen et al., 2025) to test whether verbalized rationales genuinely reflect a model’s internal decision process (Chen et al., 2025). That work manipulates the presence and semantics of hints in prompts and measures whether models both *use* and *verbalize* those hints, revealing that stated chains of thought are often unfaithful to underlying reasoning. We adopt the same causal-testing philosophy but apply it to explanation content rather than reasoning text. Specifically, the contrary hint serves as a controlled semantic intervention on the information encoded in the explanation, and the resulting change in model prediction quantifies its *fidelity*. In this formulation, faithfulness is defined as prediction sensitivity: an explanation is faithful if altering its truth value changes the model’s decision. Consequently, our contrary-hint score provides a direct, model-agnostic instantiation of the evaluation approach advocated by Chen et al. (2025), complementing prior intervention tests on CoT text (Lanham et al., 2023) by operating at the level of meaning rather than surface form.

Theorem 1 (Latent-Context Intervention Validity for Faithfulness). *Let $f : \mathcal{X} \times \mathcal{C} \to \Delta(\mathcal{Y})$ be a language model mapping an input X and latent context C to a predictive distribution over an output space $\mathcal{Y}$. Let E_{NL} denote a natural-language explanation of $f(X;C)$, and let $\neg E_{NL}$ denote its contrary hint. Assume that E_{NL} asserts a proposition about a semantic factor $S_E = s(X,C)$, where $s(\cdot)$ extracts the decision-relevant concept, which is latent or retrieved, that the explanation verbalizes. Conditioning on $\neg E_{NL}$ is equivalent to intervening on this factor while holding (X,C) fixed, i.e., $f(X;C \mid \neg E_{NL}) = f(X;C \mid do(S_E \leftarrow \bar{s}))$ for some contradictory value $\bar{s}$, and predictions are invariant to any irrelevant text R, so $f(X;C) = f(X \cup R;C)$. Defining $S_E(X;C) = D(f(X;C), f(X;C \mid \neg E_{NL}))$, where D is any strictly proper divergence, we have*

$$S_E(X;C) = 0 \iff E_{NL} \text{ is non-faithful for } f(X;C),$$
$$S_E(X;C) > 0 \iff E_{NL} \text{ is faithful for } f(X;C).$$

Hence, the contrary-hint score S_E constitutes a valid empirical estimator of faithfulness when the decision-relevant content is not contained in the observed input X but arises from latent or retrieved context.

Intuition of the theoretical foundation. Theorem 1 formalizes this causal perspective: if the explanation encodes information that lies on a causal path to the model’s output, then intervening with a contradictory hint necessarily shifts the predictive distribution, yielding a positive contrary-hint score. Conversely, if the explanation is merely correlational or decorative, the intervention leaves the model unchanged and the score remains zero. Thus, the contrary-hint fidelity measure S_E causal faithfulness by directly testing whether the model’s output is *functionally dependent* on the semantics expressed in its own explanation. More discussions and proof are in Appendix A and Corollary 1.

3.2 `FaithLM` on Fidelity-enhanced Explanation

In this section, we introduce an iterative framework designed to progressively enhance the fidelity of NL explanations. The primary goal of `FaithLM` here is to generate faithful NL explanations with iterative fidelity-enhanced optimization.

Algorithm 1 Fidelity-enhanced Explanation

```
Input: Input $\boldsymbol{X}$, output $\boldsymbol{Y}$, targeted LLMs $f(\cdot)$, human-crafted prompt $\mathcal{P}_{\mathcal{E}}$, and LLM explainer $g_E(\cdot)$.
Output: NL explanation $\mathcal{E}_{\mathrm{NL}}$.
1: $\mathcal{E}_{\mathrm{NL}} := g_E(\mathcal{P}_{\mathcal{E}} \mid \boldsymbol{X}, \boldsymbol{Y})$
2: $\mathcal{T} = \varnothing$
3: while steps not end and decision not flips do
4:     Estimate the fidelity score $\mathcal{S}_{\mathcal{E}}$ of $\mathcal{E}_{\mathrm{NL}}$
5:     Append $\mathcal{T}.append([\mathcal{E}_{\mathrm{NL}},\ \mathcal{S}_{\mathcal{E}}])$
6:     Update $\mathcal{E}_{\mathrm{NL}} := g_E(\mathcal{T}, \boldsymbol{X}, \boldsymbol{Y})$
7: end while
```

Fidelity-enhanced Explanation. The framework of fidelity-enhanced explanation is illustrated in Figure 2(a). Since the initial explanation may be unreliable and unfaithful, we propose a fidelity-enhanced optimization approach designed to progressively generate explanations with higher fidelity. We aim to explain the response $\boldsymbol{Y}$ produced by the targeted LLM $f(\cdot)$ in response to the given input queries $\boldsymbol{X}$ with NL explanations $\mathcal{E}_{\mathrm{NL}}$ following the goal of fidelity enhancement. In the first round of enhancement, the LLM explainer generates NL explanations $\mathcal{E}_{\mathrm{NL}}$ following a given human-crafted explanation trigger prompt $\mathcal{P}_{\mathcal{E}}$ provided in Appendix H. The explanations are then generated by the explainer $g_E(\mathcal{P}_{\mathcal{E}} \mid \boldsymbol{X}, \boldsymbol{Y})$. Starting from the second round till converge, FaithLM collects a trajectory $\mathcal{T}$ with the NL explanations $\mathcal{E}_{\mathrm{NL}}$ and their corresponding fidelity scores $\mathcal{S}_{\mathcal{E}}$ generated by Fidelity Evaluator. The collection process can be represented as $\mathcal{T}$ appended with $[\mathcal{E}_{\mathrm{NL}},\ \mathcal{S}_{\mathcal{E}}]$, where $\mathcal{T}$ initially starts as an empty trajectory. Following this trajectory, the LLM explainer generates new explanations with the goal of achieving higher fidelity scores in subsequent iterations. This process is guided by the system prompts detailed in Figure 14.

The trajectory $\mathcal{T}$ is continuously updated by incorporating each newly derived explanation with its assessed fidelity score until the convergence. Regardless of any given explanation trigger prompts $\mathcal{P}_{\mathcal{E}}$, FaithLM can all systematically guide the generation of NL explanations, progressively improving fidelity scores by following the reference path established in the trajectory.

Algorithm of Fidelity-enhanced Explanation. The outline of FaithLM for Fidelity-enhanced Explanation is detailed in Algorithm 1. Specifically, in the first iteration, FaithLM generates the NL explanations using the human-craft prompts (line 1). starting from the second iteration till the convergence or optimization ends, FaithLM estimates the fidelity of the derived NL explanations (line 4). Then, we incorporate the explanation and its corresponding fidelity score to the trajectory (line 5), and update the explanations with the goal of achieving higher fidelity scores in subsequent iterations (line 6). The iteration terminates at a predetermined step or ceases earlier as soon as FaithLM observes a flipping performance from the targeted LLM $f(\cdot)$.

Algorithm 2 Trigger Prompt Optimization.

```
Input: Hold-out dataset $\mathcal{D}$, Targeted LLMs $f(\cdot)$, and LLM explainers $g_E(\cdot)$.
Output: Optimal explanation trigger prompt $\mathcal{P}_{\mathcal{E}}$.
Initialize human-crafted $\mathcal{P}_{\mathcal{E}}$
Initialize $\mathcal{T} = \{\varnothing\}$
while (Steps Not End) do
    for $(\boldsymbol{X}_i, \boldsymbol{Y}_i) \sim \mathcal{D}$ do
        $\mathcal{E}_i := g_E(\mathcal{P}_{\mathcal{E}} \mid \boldsymbol{X}_i, \boldsymbol{Y}_i)$
        Estimate the fidelity score $\mathcal{S}_i$ of $\mathcal{E}_i$
    end for
    $\mathcal{S}_{\mathcal{P}} = \mathbb{E}_{\mathcal{E}_i \sim g_E(\mathcal{P}_{\mathcal{E}} \mid \boldsymbol{X}_i, \boldsymbol{Y}_i)}\left[\mathcal{S}_{\mathcal{E}_i}\right]$
    Append $\mathcal{T}.append(\mathcal{T} \cap (\mathcal{P}_{\mathcal{E}}, \mathcal{S}_{\mathcal{P}}))$
    Update $\mathcal{P}_{\mathcal{E}} := g_E(\mathcal{P}_{\mathcal{E}} \mid \mathcal{D})$
end while
```

3.3 FaithLM on Trigger Prompt Optimization

Despite the success of enhancing fidelity in Section 3.2, the low quality of the explanation trigger prompts $\mathcal{P}_{\mathcal{E}}$ may still hinder the optimization process of receiving a high-fidelity explanation. Given that the unknown preference for prompts from LLMs, human-crafted trigger prompts used in Fidelity-enhanced Explanation Optimization might lead to sub-optimal fidelity enhancement in the derived explanations. In this section, we hereby propose a new optimization pipeline under FaithLM, aiming to optimize the trigger prompt $\mathcal{P}_{\mathcal{E}}$ for generating NL explanations with higher fidelity scores as the LLM explanations of input each input query.

Trigger Prompt Optimization. The framework of Trigger Prompt Optimization is shown in Figure 2(b). The framework aims to optimize the trigger prompt to generate NL explanations with higher fidelity. Different from the optimization goal in Section 3.2, the trajectory in this task collects the trigger prompts $\mathcal{P}_{\mathcal{E}}$ and their fidelity scores $\mathcal{S}_{\mathcal{P}}$. The trajectory is constructed by the system optimization prompts detailed in Figure 13.

To estimate the fidelity score for a trigger prompt, FaithLM first adopts the randomly human-crafted trigger prompt to guide the LLM explainers to generate NL explanations, and then utilize the Fidelity Evaluator to assess the fidelity of the derived explanation. The final estimated score is averaged by the fidelity score $\mathcal{S}_{\mathcal{E}_i}$ of the hold-out dataset $(\boldsymbol{X}_i, \boldsymbol{Y}_i) \in \mathcal{D}$. Formally, the fidelity score for a trigger prompt $\mathcal{P}_{\mathcal{E}}$ is as follows:

$$\mathcal{S}_{\mathcal{P}} = \mathbb{E}_{\mathcal{E}_i \sim g_E(\mathcal{P}_{\mathcal{E}} | \boldsymbol{X}_i, \boldsymbol{Y}_i)}\left[\mathcal{S}_{\mathcal{E}_i}\right], \tag{1}$$

where $\mathcal{S}_{\mathcal{E}i}$ represents the fidelity score of the explanation $\mathcal{E}_i$, which is generated by $g_E(\mathcal{P}_{\mathcal{E}}|\boldsymbol{X}_i, \boldsymbol{Y}_i)$, as assessed by the Fidelity Evaluator.

During the optimization, the trajectory begins from an empty set and starts to incorporate newly derived trigger prompts with the fidelity scores in each optimization iteration. Following this trajectory, the LLM explainer generates a new trigger prompt with the goal of achieving higher fidelity scores of explanations in subsequent iterations. After several rounds of iterations, FaithLM ultimately yields an optimal explanation trigger prompt with the highest fidelity score for the LLM explainer to generate a more faithful NL explanation.

Algorithm of Trigger Prompt Optimization. The outline of FaithLM for Trigger Prompt Optimization is detailed in Algorithm 2, which focuses on optimizing the trigger prompt for generating NL explanations. Specifically, in each iteration, LLM explainer $g_E(\cdot)$ leverages the trigger prompt to generate the NL explanations and estimates its fidelity (lines 4-7). The fidelity scores of the trigger prompts average the fidelity scores of the entire hold-out dataset (lines 8). Afterward, the trajectory appends the trigger prompt with its corresponding fidelity score (line 9), and updates the trigger prompt as a new sequence of words to achieve higher fidelity scores (line 10). Through multiple iterations, FaithLM progressively guides the trigger prompt to generate explanations with higher fidelity scores, following the reference path established in the trajectory. The iteration process terminates at a predetermined 20 step.

4 Experiment

In this section, we conduct experiments to evaluate the performance of FaithLM, aiming to answer the following three research questions: **RQ1:** How does FaithLM perform in generating explanations in terms of efficacy? **RQ2:** Can optimized explanation trigger prompts transfer between different datasets? **RQ3:** Does the configurations of LLMs affect the explanation performance of FaithLM?

4.1 Dataset and Baseline

Datasets. We evaluate FaithLM on three datasets with multiple tasks: ECQA (Aggarwal et al., 2021) dataset on commonsense question-answer task, TrivaQA-Long (Bai et al., 2023; Joshi et al., 2017) dataset on reading comprehension task, and COPA (Kavumba et al., 2019; Roemmele et al., 2011) dataset on commonsense causal reasoning task. More details of datasets are provided in Appendix B. **Baseline Methods.** We compare FaithLM with two state-of-the-art baseline methods: SelfExp (Madsen et al., 2024) and Self-consistency (Wang et al., 2022). The former ones instruct LLMs to generate explanations using prompt engineering under single-forward inference, and the later ones leverage the outputs from the chain-of-thought prompting process as the model explanations.

Comparison with Concurrent Faithfulness Notions. Simulatability-based evaluations (e.g., (Madsen et al., 2024; Chen et al., 2023a)) assess whether a model can reproduce predictions from an explanation, i.e., behavioral imitation. In contrast, FaithLM measures *causal sensitivity* via a controlled intervention on the explanation's semantics: we compute $S_E = f(X) - f(X \mid \neg E_{NL})$ and read off prediction shifts induced by a contrary hint, without token masking or task-specific heuristics. This aligns with intervention-driven faithfulness testing advocated by recent work and keeps the target model f unchanged.[1] Empirically, we additionally report a small-scale comparison where we compute (i) simulatability scores and (ii) S_E on the same instances; we analyze where the two agree and where S_E detects causal use beyond imitability.

4.2 Experimental Settings

We evaluate FaithLM through two complementary tasks: (i) fidelity-enhanced explanation generation and (ii) explanation-trigger prompt optimization. Details are presented in Appendix C.

Fidelity-Enhanced Explanation. We measure the alignment between generated explanations and predictions (Chen et al., 2025). For each example, FaithLM produces one NL explanation, and the average fidelity score across instances is reported as the final metric.

Explanation-Trigger Prompt Optimization. This task optimizes the trigger prompt that guides FaithLM to generate more faithful explanations. At each iteration, 30 samples are drawn from the training data as a validation set, and the mean fidelity score on these samples is used as the optimization objective.

Evaluation Metrics. We evaluate explanation quality using two metrics: *fidelity* and *truthfulness*. Fidelity follows the intervention-based protocol of Chen et al. (2025), where contrary hints act as controlled semantic interventions and fidelity is the resulting prediction sensitivity. Truthfulness measures semantic consistency between generated and ground-truth explanations using GPT-4o, RoBERTa-Large, and XLNet-Large (Nie et al., 2020), following Liu et al. (2023). Evaluators classify each pair as *similar*, *dissimilar*, or *non-relevant*, and the proportion of "similar" outputs forms the truthfulness score. Full prompts appear in Appendix I.

Implementation Details. We use Vicuna-7B (Chiang et al., 2023) and Phi-2 (Javaheripi and Bubeck, 2023) as target models $f(\cdot)$, and GPT-3.5-Turbo and Claude-2 (Anthropic, 2023) as explainers $g_E(\cdot)$. The same models generate contrary hints. Results are averaged over three runs with grid search on hyperparameters. Both Vicuna-7B and Phi-2 share identical decoding settings. Full configurations and infrastructure details are given in Appendices E and F.

4.3 Explanation Efficacy of FaithLM (RQ1)

Efficacy of Derived Explanations. We assess the efficacy of derived explanations under the fidelity metric. FaithLM adopts the trajectory system prompts in Figure 14 of Appendix J. The generation of contrary hints is guided by the prompt in Table 8.

- **Fidelity Evaluation.** The results in Figure 3 demonstrate that FaithLM achieves significantly higher fidelity scores across all three datasets compared with

[1] See our definition and evaluator description in §3.1.

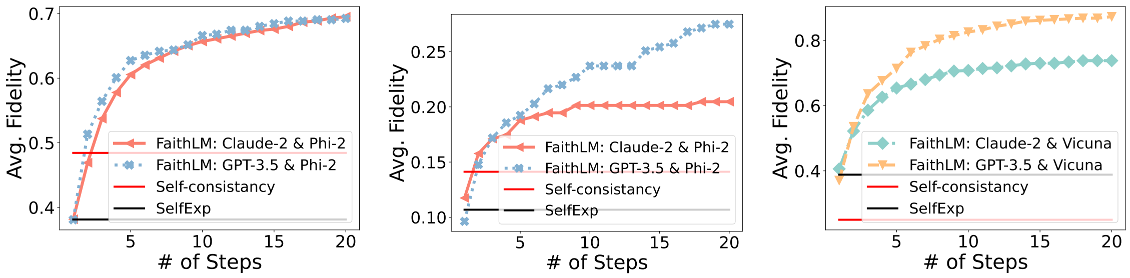


Figure 3: The fidelity evaluation of explanations on ECQA (left), TriviaQA-Long (middle), and COPA dataset (right). The reported scores are the average fidelity on testing instances in each step of fidelity-enhanced optimization.

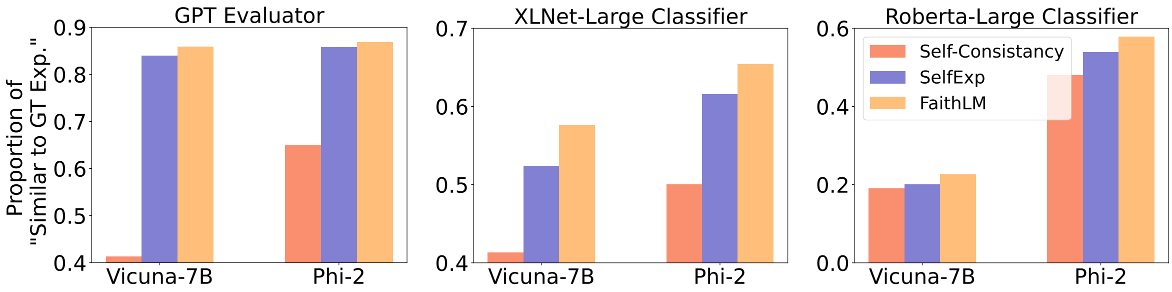


Figure 4: Trustfulness evaluation of the NL explanations. Higher the proportion of "similar to ground-truth explanation," the more consistent the derived explanations are with the ground-truth NL explanations.

two baselines after 20 steps of optimization. Moreover, the optimization curve of fidelity demonstrates that 20 rounds of optimization are sufficient to converge. A similar phenomenon occurs across different settings of explainers and targeted LLMs. Additional results are provided in Appendix G.

- **Truthfulness Evaluation.** To evaluate the truthfulness of explanations, we show the proportions of "similar to ground-truth explanations" in the ECQA dataset, as depicted in Figure 4. We leverage GPT evaluators and well-trained NLI evaluators to assess whether the given explanations are within similar content to ground-truth explanations. The results show that FaithLM's explanations are more consistent with the ground-truth NL explanations, indicated by a larger proportion of "similar to ground-truth explanations" generated by FaithLM than baseline methods.

Efficacy of Explanation Trigger Prompts. We first show prompt optimization curves on three different datasets, and then leverage the optimal explanation trigger prompts to generate explanations via FaithLM. In the experiments, we randomly select 15 instances from the training dataset in each optimization round, and compute the average fidelity scores of the newly derived trigger prompts. After the progress is terminated, we evaluate the optimized trigger prompts on the testing set. The optimization step is uniformly established at 50 rounds across different explainer and targeted LLMs.

- **Trigger Prompt Optimization Curve.** Figure 5 demonstrates the optimization curves of three datasets. We display the explainer as GPT-3.5-Turbo and Claude-2 and the explainer as Vicuna-7B. We observe that the optimization curve exhibits a generally ascending trend as the step progresses, interspersed with multiple waves throughout the optimization procedure. This indicates that FaithLM generates better explanation trigger prompts after the optimization. More results of optimization curves on remaining datasets are provided in Appendix G.

- **Explanation Generation by Optimized Trigger Prompts.** We utilize the optimized explanation trigger prompts to generate explanations following Algorithm 1. The results are displayed in Figure 6(a), including the experiments conducted using all three datasets with Claude-2 as the explainer and Vicuna-7B as the targeted LLM. We observe that optimized explanation trigger prompts obtain higher fidelity scores than the initial human-crafted trigger prompt in generating explanations. This trend is consistent across all datasets, regardless of whether the explanations are refined by Algorithm 1.

4.4 Transferability of Trigger Prompt (RQ2)

We assess the transferability of ultimately optimized trigger prompts across different unseen datasets within the same domain, as depicted in Figure 6(b). Specifically, we transfer the optimized trigger prompts from the ECQA to the Social-IQA dataset, and from the COPA to the XCOPA datasets, without any additional optimization. Specifically, the Social-IQA dataset is dedicated to commonsense question-answering (similar to the ECQA dataset), while the XCOPA dataset specializes in causal reasoning (similar to the COPA dataset). We adopt the Vicuna-7B as the targeted LLM, and Claude-2 as the

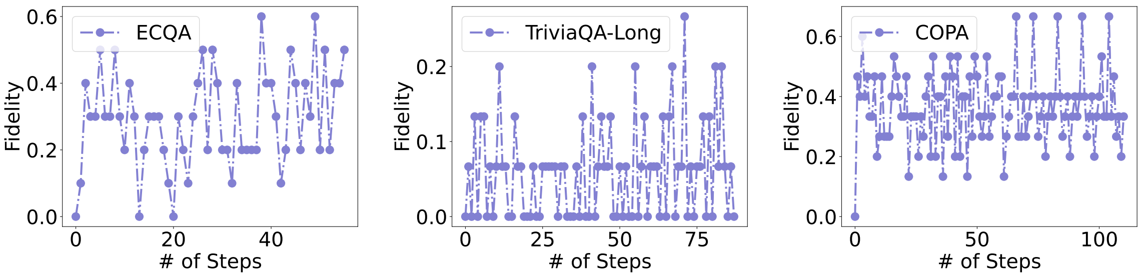
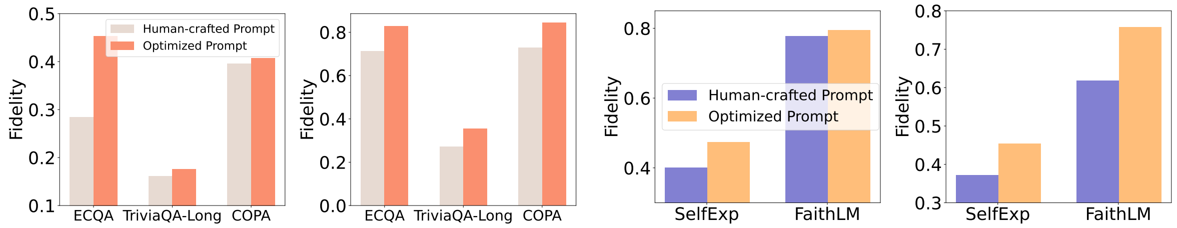


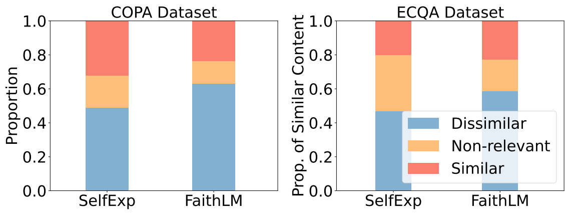
Figure 5: The fidelity in different optimization steps of the trigger prompts (Algorithm 2) on the ECQA, TrivaQA, and COPA datasets. The fidelity grows higher as the number of steps increases.



(a) Robustness analytics of trigger prompts

(b) Transferability results of trigger prompts

Figure 6: Assessment on the adaptation of the optimized explanation trigger prompts. Figure (a) reveals the robustness evaluation, and figure (b) illustrates the results on transferability.



(a) Evaluated by GPT-4o Evaluator

Figure 7: Ablation studies on evaluating contrary hint.

explainer on these transfer tasks. The fidelity of the derived NL explanation on the target dataset is shown in Figure 6(b). The optimized trigger prompts show better explanation efficacy than human-crafted prompts when it is transferred in similar domain. This shows that the optimized trigger prompts generated by FaithLM possess a great data transferability.

4.5 Ablation Studies on contrary hint (RQ3)

The quality of contrary hints $\neg\mathcal{E}_{\text{NL}}$ determines the efficacy of FaithLM. We leverage the powerful LLMs as the LLM agent to generate contrary hints, requesting the delivery of high-quality opposite-meaning outputs from their original NL explanations. In this section, we evaluate the quality of contrary hints, aiming to observe the semantic differences between the original NL explanations and their contrary hints. To examine the quality, we employ one GPT-4o classifier and two well-trained NLI classifiers, Roberta-Large and XLNet-Large (Nie et al., 2020). We leverage each classifier to distinguish whether the relationship between the “original NL explanations" and “contrary hints" belong to the category of “similar meaning (entailment)," “dissimilar meaning (contradiction)," or “non-relevant (neutral)." We follow the evaluation settings from (Liu et al., 2023) on GPT-classifier with evaluation prompt provided in Table 9.

The results are shown in Figure 7 and Figure 11 under the randomly sampled 100 instances from the ECQA and COPA datasets. We observe that the two NLI classifiers achieve up to 86% and 82% in the “dissimilar meanings"" category on the ECQA and COPA datasets, respectively. The results of the GPT-classifier demonstrate that the derived explanations from SelfExp obtain more non-faithful information than FaithLM, risking the LLM agent of Fidelity Evaluator in generating non-relevant information as the contrary hints. Case studies are provided in Appendix K to show the informativeness and readability of contrary hints.

5 Conclusion

In this paper, we introduce FaithLM to explain the decision-making process of LLMs, instead of providing reasoning or self-refinement feedback as model explanation. Specifically, FaithLM employs a fidelity enhancement strategy to progressively refine the fidelity of derived explanations and explanation trigger prompts. FaithLM conducts an iterative process to improve the fidelity of derived explanations. Theorem 1 establishes the contrary-hint score as a valid measure of faithfulness. Experimental results demonstrate the effectiveness of FaithLM, and better alignment with the ground-truth explanations. This suggest that the decision-making process are truly reflected. For future work, we plan to extend FaithLM in healthcare, where the needs for transparency is critical given the growing reliance on black-box LLMs.

6 Limitations

One significant limitation of FaithLM associated with the carbon emissions during the experiments. To generate better fidelity and truthfulness of the derived explanations, FaithLM requires iterating a few rounds of optimization during the explanation derivation. This leads to extra computational resources to proceed. The extra computational power is thus required for optimizing FaithLM leads to considerable energy consumption, which, in turn, results in a significant carbon footprint. As the demand for more sophisticated LLMs continues to grow, so does their environmental impact. This limitation underscores the urgent need to explore and adopt more sustainable practices and technologies in the development and learning FaithLM with fewer optimization steps to mitigate their ecological footprint.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, page 3050–3065. Association for Computational Linguistics.

Anthropic. 2023. Claude 2.

AnthropicAI. 2023. Introducing claude. *See https://www.anthropic.com/index/introducing-claude*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. 2023a. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, pages 1–12.

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021. Kace: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516–2527.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. 2025. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2023b. Do models explain themselves? counterfactual simulatability of natural language explanations. *arXiv preprint arXiv:2307.08678*.

Zichen Chen, Ambuj K Singh, and Misha Sra. 2023c. Lmexplainer: a knowledge-enhanced explainer for language models. *arXiv preprint arXiv:2303.16537*.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training. *arXiv preprint arXiv:2311.04155*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yu-Neng Chuang, Guanchu Wang, Fan Yang, Zirui Liu, Xuanting Cai, Mengnan Du, and Xia Hu. 2023. Efficient xai techniques: A taxonomic survey. *arXiv preprint arXiv:2302.03225*.

Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Luciano Floridi. 2019. Establishing the rules for building trustworthy ai. *Nature Machine Intelligence*, 1(6):261–262.

Bryce Goodman, Seth Flaxman, and Y X. 2017. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. 2023. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*.

Mojan Javaheripi and Sébastien Bubeck. 2023. Phi-2: The surprising power of small language models.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China. Association for Computational Linguistics.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*.

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*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Gianluigi Lopardo, Frederic Precioso, and Damien Garreau. 2023. Faithful and robust local interpretability for textual predictions. *arXiv preprint arXiv:2311.01605*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.

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

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337, Bangkok, Thailand. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Oana-Maria Camburu, Thomas Lukasiewicz, and Julian McAuley. 2021. Knowledge-grounded self-rationalization via extractive and natural language explanations. *arXiv preprint arXiv:2106.13876*.

Ramesh Manuvinakurike, Emanuel Moss, Elizabeth Anne Watkins, Saurav Sahay, Giuseppe Raffa, and Lama Nachman. 2025. Thoughts without thinking: Reconsidering the explanatory value of chain-of-thought reasoning in llms through agentic pipelines. *arXiv preprint arXiv:2505.00875*.

Rakesh R Menon, Kerem Zaman, and Shashank Srivastava. 2023. Mantle: Model-agnostic natural language explainer. *arXiv preprint arXiv:2305.12995*.

Miquel Miró-Nicolau, Antoni Jaume-i Capó, and Gabriel Moyà-Alcover. 2024. A comprehensive study on fidelity metrics for xai. *arXiv preprint arXiv:2401.10640*.

Christoph Molnar. 2022. Interpretable machine learning: A guide for making black box models explainable.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, et al. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Noah Y 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. *arXiv preprint arXiv:2404.03189*.

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

Sree Harsha Tanneru, Dan Ley, Chirag Agarwal, and Himabindu Lakkaraju. 2024. On the hardness of faithful chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2406.10625*.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. *arXiv preprint arXiv:2309.10621*.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of llms via imagination, searching, and criticizing. *arXiv preprint arXiv:2404.12253*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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. *arXiv preprint arXiv:2305.04388*.

Guanchu Wang, Yu-Neng Chuang, Fan Yang, Mengnan Du, Chia-Yuan Chang, Shaochen Zhong, Zirui Liu, Zhaozhuo Xu, Kaixiong Zhou, Xuanting Cai, et al. 2023. Leta: Learning transferable attribution for generic vision explainer. *arXiv preprint arXiv:2312.15359*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *arXiv preprint arXiv:2309.01029*.

Appendix

A Proof of Theorem

Theorem 1 (Latent-Context Intervention Validity for Faithfulness). *Let $f: \mathcal{X} \times \mathcal{C} \to \Delta(\mathcal{Y})$ be a language model mapping an input X and latent context C to a predictive distribution over an output space $\mathcal{Y}$. Let E_{NL} denote a natural-language explanation of $f(X;C)$, and let $\neg E_{NL}$ denote its contrary hint. Assume that E_{NL} asserts a proposition about a semantic factor $S_E = s(X,C)$, where $s(\cdot)$ extracts the decision-relevant concept, which is latent or retrieved, that the explanation verbalizes. Conditioning on $\neg E_{NL}$ is equivalent to intervening on this factor while holding (X,C) fixed, i.e., $f(X;C \mid \neg E_{NL}) = f(X;C \mid do(S_E \leftarrow \bar{s}))$ for some contradictory value $\bar{s}$, and predictions are invariant to any irrelevant text R, so $f(X;C) = f(X \cup R; C)$. Defining $S_E(X;C) = D(f(X;C), f(X;C \mid \neg E_{NL}))$, where D is any strictly proper divergence, we have*

$$S_E(X;C) = 0 \iff E_{NL} \text{ is non-faithful for } f(X;C),$$
$$S_E(X;C) > 0 \iff E_{NL} \text{ is faithful for } f(X;C).$$

Hence, the contrary-hint score S_E constitutes a valid empirical estimator of causal faithfulness even when the decision-relevant content is not contained in the observed input X but arises from latent or retrieved context.

Proof. By the causal definition of faithfulness, an explanation is faithful if and only if altering the truth value of the decision-relevant semantic factor changes the model's prediction while holding other causes fixed. Because E_{NL} asserts a proposition about $S_E = s(X,C)$, and conditioning on $\neg E_{NL}$ implements $do(S_E \leftarrow \bar{s})$ with (X,C) fixed, we have $f(X;C \mid \neg E_{NL}) = f\big(X;C \,\big|\, do(S_E \leftarrow \bar{s})\big)$. If E_{NL} is non-faithful, then S_E has no causal influence on Y under (X,C) and the intervention leaves the predictive distribution unchanged, yielding $S_E(X;C) = 0$. If E_{NL} is faithful, S_E lies on a causal path to Y under (X,C) and the intervention changes the predictive distribution; strict propriety of D implies $S_E(X;C) > 0$. Invariance under irrelevant spans follows by the stated stability condition and applies to both terms inside D. □

Corollary 1 (Robustness and Monotonicity). *Under contextual stability, the contrary-hint score remains invariant to irrelevant input variations, satisfying*

$$S_E(X;C) = S_E(X \cup R; C) \quad \text{for any semantically irrelevant } R.$$

Furthermore, if an iterative procedure produces a sequence of explanations $\{E_{NL}^{(t)}\}$ such that $S_E^{(t+1)}(X;C) \geq S_E^{(t)}(X;C)$ for all iterations prior to convergence or a decision flip, then the sequence is non-decreasing in causal faithfulness. Consequently, any iteration with $S_E^{(t)}(X;C) > 0$ corresponds to a faithful explanation.

B Details about Datasets

The experiments are conducted on the three NLU datasets. The details of the datasets are provided as follows:

- **ECQA (Aggarwal et al., 2021).** ECQA is an extension of the CQA dataset (Talmor et al., 2019). Specifically, based on the CQA dataset, it annotates the positive or negative properties and golden explanations for the QA pairs. Due to API cost budgets, we evaluate our framework on the first 500 instances in the ECQA dataset.
- **TriviaQA LongBench (Joshi et al., 2017).** TriviaQA LongBench (TriviaQA-Long) is a reading comprehension dataset. It includes 300 question-answer-evidence triples sourced from the Longbench (Bai et al., 2023) dataset[2]. This dataset features question-answer pairs crafted by trivia enthusiasts, accompanied by independently sourced evidence documents, providing supervision for answering these questions.
- **Balanced COPA (Roemmele et al., 2011; Kavumba et al., 2019).** The Balanced COPA (COPA) dataset is a collection of 500 questions for commonsense causal reasoning. Each question consists of a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise.

C Experiment Settings

We introduce the experimental settings for evaluating `FaithLM`. Two distinct types of explanation tasks and evaluation settings are as follows.

Fidelity-enhanced Explanation In this task, our goal is to produce NL explanations that exhibit a higher fidelity. The fidelity is exploited as a metric to evaluate fidelity. `FaithLM` is evaluated across all testing instances, where an NL explanation is generated for each instance, and the averaged fidelity score is calculated, serving as the reported metric to evaluate fidelity.

[2] https://huggingface.co/datasets/THUDM/LongBench/

Explanation Trigger Prompt Optimization. In this task, we aim to optimize the explanation trigger prompt that benefits `FaithLM` in generating better explanations. The optimization process is conducted on the same dataset, where 30 instances are sampled as a hold-off dataset in each optimization step from the training set. During the optimization process, the fidelity score of a trigger prompt is calculated as the average of the fidelity scores from the selected instances.

Evaluation Metrics. The quality of the derived NL explanation is evaluated under the fidelity and truthfulness metrics. The fidelity follows the prior work (Chen et al., 2025), which observes the discrepancy of the targeted LLMs by incorporating contrary hints to the input. The evaluation of truthfulness assesses the correlation between the derived NL explanations to the ground-truth explanations Specifically, we leverage GPT-4o and two well-trained natural language inference (NLI) models, Roberta-Large and XLNet-Large (Nie et al., 2020) from the huggingface hub (Wolf et al., 2019), as the evaluators. With the same evaluators setup, the truthfulness evaluation follows the settings from (Liu et al., 2023), and uses the evaluation prompt provided in Appendix I. Specifically, the evaluators assess the derived explanations and ground-truth explanations, determining whether the two sentences belong to "similar content", "dissimilar content," or "non-relevant content". Higher the proportion of "similar content", the more consistent results with ground-truth NL explanations.

Implementation Details. In the experiments, we explore two variants of LLMs as the targeted LLMs $f(\cdot)$: Vicuna-7B (Chiang et al., 2023) and Phi-2 (Javaheripi and Bubeck, 2023), two types of LLMs as the explainers $g_E(\cdot)$ in `FaithLM`: GPT-3.5-Turbo and Claude-2 (Anthropic, 2023). The LLM agent for generating the contrary hints takes the same LLMs as those used by the explainers. All reported results are calculated from the average scores of 3 times repetitions with the grid search on the performance. The settings for predictors are uniform, with Phi-2 (2.7B) and Viucua-7B receiving identical hyperparameter configurations during the experiments conducted in this study.

D Related Work

D.1 Post-hoc Explanation

Post-hoc explanation techniques have undergone significant development and discussion, driven by the widespread adoption of black-box ML models across various data modalities. A multitude of post-hoc algorithms has been introduced from two aspects: local and global explanations (Molnar, 2022; Du et al., 2019). Explanations aim to explain the reasoning behind an individual model for each input instance, while global explanations aim to uncover the overall functioning of a complex model (Chuang et al., 2023). Considering various purposes of explanation, the explanation techniques mainly showcase the explanation from two perspectives, including feature attributions and counterfactual examples. Feature attribution aims to provide users with important scores for each feature's impact on model predictions, while counterfactual examples aim to offer alternative instances that explicitly assist users in grasping the model's decision-making process. In recent years, with the growing proficiency and wide usage of black-box LLMs, especially closed-source LLMs service, post-hoc explanations have become increasingly prominent and have garnered significant attention in NLP research due to the inaccessibility of LLMs' model weights and structure (Zhao et al., 2023).

D.2 Explainability of LLMs

The majority of explanation efforts in LLM research have centered on delivering explanations. One group of studies calculates importance scores for specific tokens (Lopardo et al., 2023; Huang et al., 2023), another line of progress generates NL explanations by leveraging the pre-trained LLMs with internal model knowledge sources (Kumar and Talukdar, 2020; Chen et al., 2023c; Menon et al., 2023), the other group of work leverages LLMs themselves to generate chain-of-thought (CoT) reasoning (Lanham et al., 2023; Radhakrishnan et al., 2023; Chen et al., 2023b,b) as the self-explanations through the one feed-forward inference process. Furthermore, some studies aim to yield counterfactual explanations by pre-trained LLMs to assist users in better understanding the decision-making process from LLMs (Chen et al., 2021, 2023b). Although NL explanations offer fantastic human-understandable insights than token-wise explanations, the explanations can lose their fidelity via one feed-forward inference process of pre-trained LLMs. Unreliability and non-fidelity of NL explanations are still a concern (Ye and Durrett, 2022; Turpin et al., 2023). Given our primary aim of producing faithful explanations, our efforts are to generate NL explanations to improve the likelihood of accurately representing the decision-making process of LLMs.

D.3 LLMs as Optimizers

LLMs as optimizers is a novel paradigm, describing optimization problems in natural language and utilizing the reasoning capabilities of LLMs for optimizing (Yang et al., 2023). Depicting optimization problems in natural language enables the optimization of diverse tasks without defining formal specifications, such as prompt optimization (Yang et al., 2023; Cheng et al., 2023; Guo et al., 2023), agent learning (Shinn et al., 2023), and model labeling (Thomas et al., 2023). Based on this optimization paradigm, our work introduces a generative explanation framework with a novel estimation method of sentence-level fidelity.

E Hyper-parameter Settings of `FaithLM`

The hyper-parameters of `FaithLM` are given in Table 2. The configuration for explainers is consistent across Claude-2 and GPT-3.5-Turbo, provided that the parameters are adjustable. Likewise, the settings for predictors are uniform, with Phi-2 and Viucua-7B receiving identical hyperparameter configurations during the experiments conducted in this study.

	Dataset	ECQA	TriviaQA-Long	COPA
Fidelity-enhanced Optimization	Optimization Steps	20	20	20
	Temperature of Predictor LLMs	0.7	0.5	0.7
	Temperature of Explainer LLMs	0.9	0.9	0.9
	Top-P of Explainer LLMs	0.9	0.9	0.9
Trigger-oriented Optimization	Optimization Steps	50	100	100
	Sampled Instances	30	30	30
	Temperature of Predictor LLMs	0.7	0.5	0.7
	Temperature of Explainer LLMs	0.9	0.9	0.9
	Top-P of Explainer LLMs	0.9	0.9	0.9

Table 2: Hyper-parameters and optimization settings in `FaithLM`.

F Computation Infrastructure and Costs

F.1 Computation Infrastructure

For a fair comparison of testing algorithmic throughput, the experiments are conducted based on the following physical computing infrastructure in Table 3.

Device Attribute	Value
Computing infrastructure	GPU
GPU model	Nvidia-A40
GPU number	1
GPU Memory	46068 MB

Table 3: Computing infrastructure for the experiments.

F.2 Computation Costs

The computational costs associated with `FaithLM` primarily differ from the inference costs of local LLMs and the expenses related to API-accessed LLMs. The computational costs depend on the parameter scale and variants of LLMs used in the `FaithLM` framework, shown in Table 4 and 5.

	ECQA	TrivaQA	COPA
Execution Time (Sec.)	∼3	∼5	∼3
Execution Cost ($)	∼0.01	∼0.04	∼0.01

Table 4: Computing costs of `FaithLM` with GPT-3.5 on each dataset.

	bs=32	bs=64	bs=96
Execution Time (Sec.)	∼3	∼5	∼3
Memory Cost (GB)	∼28GB	∼43GB	∼59GB

Table 5: Computing costs of `FaithLM` with Vicuna-7B under different batch size (bs).

G Additional Experimental results of **FaithLM**

G.1 Optimization Procedure of derived explanations

We demonstrate more evaluation results on derived explanations from FaithLM. The outcomes depicted in Figure 8 reveal that FaithLM attains notably higher fidelity scores across all three datasets following 20 steps of optimization. Additionally, Figure 8 illustrates the evolution of the optimization process during the generation of explanations.

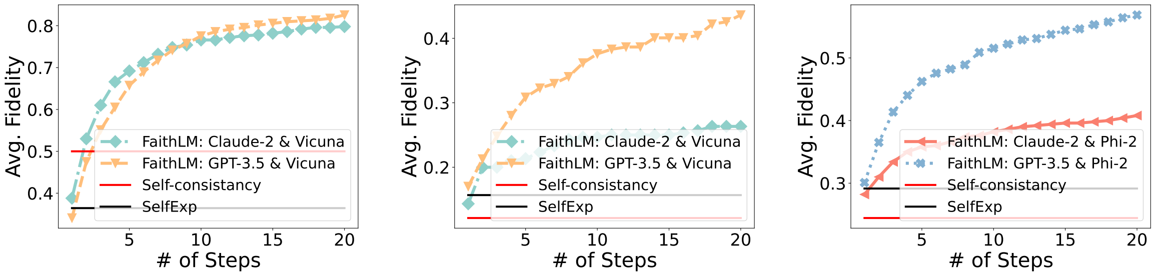


Figure 8: The fidelity evaluation of derived explanations from FaithLM under different settings of predictors and explainers.

G.2 Additional Optimization Curve of Explanation Trigger Prompt

We demonstrate more evaluation results on the optimization curve of explanation trigger prompts of FaithLM. The optimization curve shown in Figure 9 generally displays an upward trend with the progression of steps, interspersed with several fluctuations throughout the optimization process. This suggests that FaithLM can successfully generate improved explanation trigger prompts after optimization.

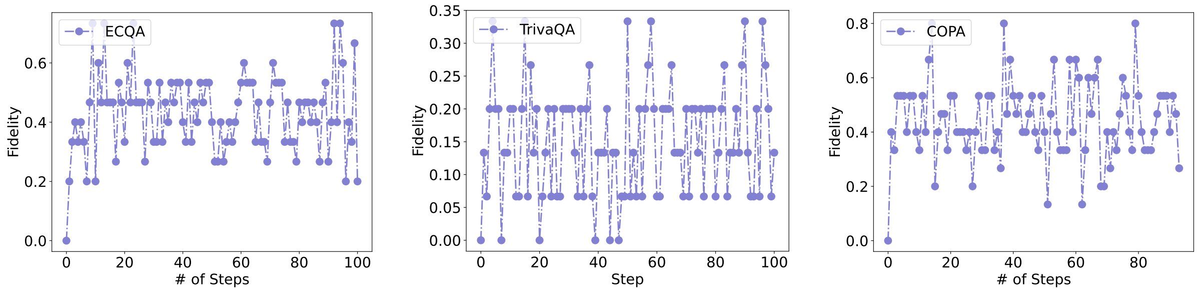


Figure 9: The optimization curve of explanation trigger prompts on ECQA (left), TriviaQA-Long (middle), and COPA dataset (right).

G.3 Additional Experiments on Diverse Domains of Dataset

We further have conducted additional experiments on one new MedMCQA dataset (Pal et al., 2022) in the healthcare domain. We evaluate the FaithLM using a fidelity assessment under Natural Language Explanation Generation settings. All experimental configurations follow the settings in Section 4.2. The experimental results are shown in the table below. We observe that FaithLM outperforms the baseline method, which is consistent with the experimental results across other domain datasets that were evaluated in our work.

	SelfExp	Self-consistency	FaithLM
Fidelity	0.6956	0.4715	**0.9565**

Table 6: Additional experimental results on MedMCQA dataset.

G.4 Additional Experiments on Truthfulness Evaluation

We evaluate all baseline methods and `FaithLM` under multiple settings to assess how closely the derived explanations match the ground-truth rationales. A GPT-based evaluator assigns a GPT-Score from 1 to 5, with higher values indicating greater semantic similarity. Explanations tagged as "similar content" or scoring near 5 are treated as matches. We report the truthfulness of generated explanations in Figure 10. A GPT-based evaluator assigns a truthfulness score from 1 to 5 by checking factual consistency with the input, the task label, and commonsense knowledge. Explanations tagged as factually consistent or scoring near 5 are counted as truthful. Across all settings, `FaithLM` attains the highest mean truthfulness score and the largest fraction of truthful explanations, indicating that our optimization improves not only fidelity but also factual quality of the produced rationales.

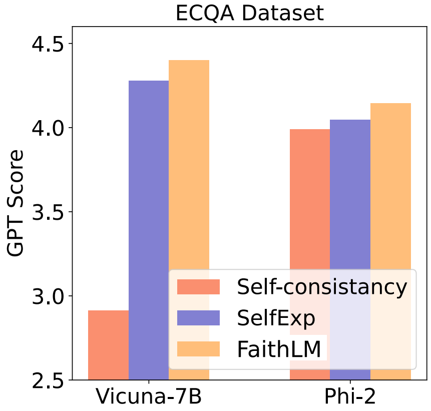


Figure 10: Truthfulness evaluation with ground-truth explanation under GPT-score settings.

G.5 Additional Experiments of Contrary Hints.

We here showcase the results of using NLI classifiers to evaluate the quality of contrary hints.

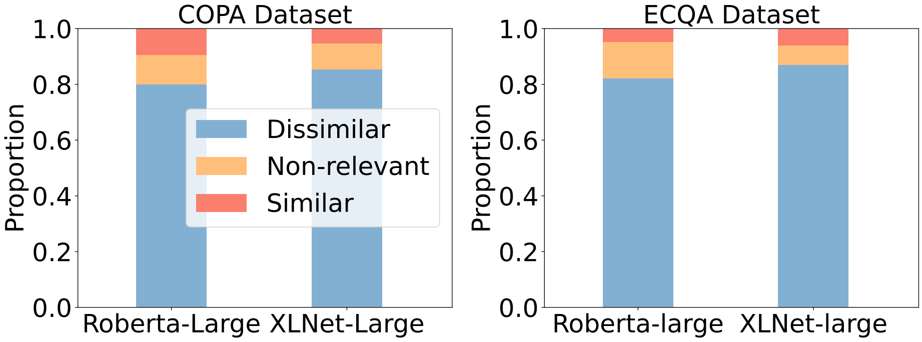


Figure 11: Quality Evaluation results of Contrary Hints via NLI classifiers

G.6 Additional Experiments of Larger Target LLM

To broaden empirical coverage beyond $\leq$7B open-source models, we additionally evaluate FaithLM on **GPT-3.5** and **Claude-2** (both $\geq$70B parameters), which differ substantially in architecture and alignment. As shown in Table 7, FaithLM consistently improves faithfulness across datasets and model families. These results indicate that the underlying causal criterion generalizes beyond the specific experimental settings of the main paper. While our evaluation focuses on MCQ-based reasoning, the formulation itself is task- and model-agnostic. Extending FaithLM to open-ended generation and multilingual domains is left for future work.

Model	Dataset	Init Fidelity	Optimized Fidelity	Δ Fidelity
GPT-3.5	ECQA	0.200	0.733	+0.533
GPT-3.5	COPA	0.400	0.800	+0.400
Claude-2	ECQA	0.600	0.667	+0.066
Claude-2	COPA	0.433	0.633	+0.200

Table 7: Faithfulness evaluation on GPT-3.5 and Claude-2 as target models.

G.7 Robustness Analytics of Configuration

In this section, the robustness test of the explainer LLMs is conducted under the analytics of hyper-parameters that are highly dependent on the outputs of LLMs. We focus on two different hyper-parameters: Temperature and Top-p. The experiments are conducted under the explainer GPT-3.5-Turbo and the predictor Vicuna-7B. We evaluate the following temperatures and top-p of the explainer LLM in the range of $\{0.3, 0.6, 0.9\}$. The results are shown in Figure 12. We observe that the explainer LLMs perform inferior when the temperature and Top-p are low, reflecting that the lower exploration of explainer LLM may degrade the optimization ability in explanation generation. The explainer LLMs are encouraged to obtain the temperatures and top-p around 0.9. The small values of the temperatures and top-p may lead to low flexibility in updating new explanations. In contrast, large temperatures and top-p may impact explainer LLMs disobeying the given optimization trajectory. Thus, in the main experiments, all reported performances are under the settings of temperature 0.9 and top-p 0.9, achieving the best performance for generating explanations.

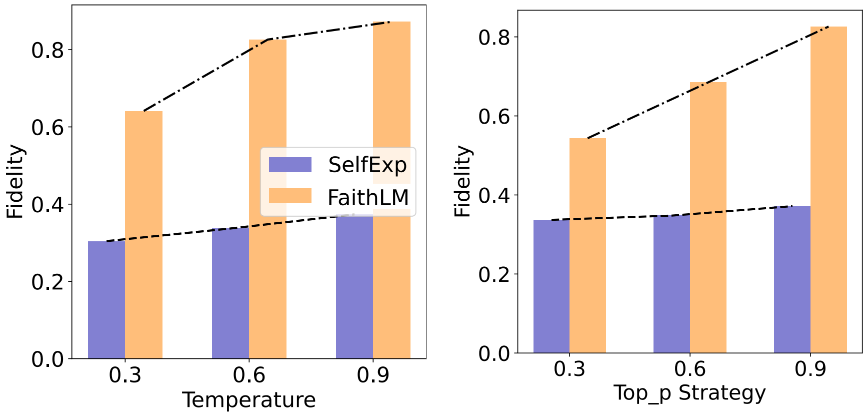


Figure 12: Robustness Analytics of `FaithLM`: Temperature (left) and Top-p strategies (right).

H Details of Prompts Usage in **FaithLM**

We provide a listing of the prompts in Table 9 utilized in FaithLM in different tasks. The first row demonstrates the initial explanation trigger prompt leveraging in both fidelity-enhanced optimization and trigger-oriented optimization. The second row shows the prompt for the LLM agent to generate the contrary hints.

Conducted Task	Evaluation Prompts
Explanation Generation	Please provide objective explanations of why the model generates the answers to the given questions based on your thoughts. Explain the reason why the model provides the answer, no matter if it is wrong or correct. Make sure not to answer the questions or provide any suggestions to better answer the questions by yourself. **Q:{*Question*}. A:{*Targeted LLM-generated Answer*}.**
contrary hints Generation	Please generate one example of obtaining the opposite meaning from a given sentence. Make sure you output sentences only. **Sentences:{*derived explanation*}.**

Table 8: The example of prompts that are given to two explainer LLMs and LLM agent for contrary hint.

I Details of Evaluation Prompt Usage

We provide a listing of the evaluation prompts in Table 9 utilized in assessing the performance of FaithLM. The first row reveals the evaluation prompt on comparing the derived explanation with the ground-truth (GT) explanation in the ECQA dataset in Section 4.3; and the second row demonstrates the evaluation prompt on activating the GPT classifier and the GPT scorer for assessing contrary hints in Section 4.5.

Evaluation Task	Evaluation Prompts
Ground-truth Explanation	Given a user instruction and two AI assistant responses, your job is to classify whether the relation of two responses in S1 and S2 belongs to G-1, G-2, or G-3. The meaning of class is as follows: (G-1) relevant contents, (G-2) irrelevant contents, or (G-3) irrelevant contents. Judge responses holistically, paying special attention to whether two responses have similar contents. Judge responses with only ONE class label as your final answer. **S1:{*derived explanation*}. S2:{*GT-Explanation*}.** Please ONLY response your in either G-1, G-2, or G-3; THERE SHOULD BE NO OTHER CONTENT INCLUDED IN YOUR RESPONSE.
GPT classifier for contrary hints	Given a user instruction and two AI assistant responses, your job is to classify whether two responses in S1 and S2 belong to G-1, G-2, or G-3. The meaning of class is as follows: (G-1) same semantic meaning, (G-2) opposite semantic meaning, and (G-3) no relation. Judge responses holistically, paying special attention to whether two responses have the same semantic meaning. Judge responses with only ONE class label as your final answer. **S1:{*derived explanation*}. S2:{*contrary hints*}.** Please ONLY respond in either G-1, G-2, or G-3; THERE SHOULD BE NO OTHER CONTENT INCLUDED IN YOUR RESPONSE.
GPT scorer of contrary hints	Given a user instruction and two AI assistant responses, your job is to rate from ONE to FIVE to judge whether two responses in S1 and S2 have the same semantic meaning or not. A FIVE score refers to being totally the same, and ONE score refers to being totally the opposite. Judge responses holistically, paying special attention to whether two responses have the same semantic meaning. The judge responds with the rates between ONE and FIVE. **S1:{*derived explanation*}. S2:{*contrary hints*}.** Please ONLY respond to the rate value; THERE SHOULD BE NO OTHER CONTENT INCLUDED IN YOUR RESPONSE.

Table 9: Evaluation Prompts given to GPT-3.5-Turbo used in assessing the efficacy of FaithLM.

J Trajectory System Prompts Usage in FaithLM

We present a detailed listing of the trigger-oriented trajectory prompt in Figure 13 and the explanation-oriented prompt in Figure 14, as utilized within the FaithLM framework.

J.1 Trigger-oriented Trajectory Prompt

System instruction: Your task is to generate the general prompts <INS> for language model generating model explanations of each question. Below are some previous prompts with their scores in the Inputs. The score is calculated as the flipping answer rates and ranges from 0 to 1.

Inputs: The following exemplars show how to apply your text:
Text: Please provide objective explanations of why model generates the answers.
Score: 0.21

Text: Provide a concise, objective explanation of only the key reasoning or assumptions that likely led the model to generate this specific response.
Score: 0.53

. . . . . .

Trajectory Instruction: Generate a prompt <INS> that is different from all prompt <INS> in Inputs above and has a higher score than all the prompts <INS> from Inputs. The prompts should begin with <INS> and end with <INS> and follow the format of the examples in Inputs. The prompts should be concise, effective, and generally applicable to all problems above.

Response: <A Newly Generated Trigger Prompt>

Figure 13: A examples of **trigger-oriented trajectory prompt**. This prompt populates in both LLM explainers, which are Cluade2 and GPT-3.5-Turbo. The output of FaithLM optimized under trigger-oriented trajectory prompt is append after the **Response** label.

J.2 NL Explanation-oriented Trajectory Prompt

System instruction: You have some texts along with their corresponding scores. The texts are the possible explanation of the following given question and answer. The texts are arranged in random order based on their scores, where higher scores indicate better quality. The scores are calculated as how relative the texts are toward the given question and answer as the explanation. The scores range from 0 to 1 based on your output text.

Inputs: The following exemplars show how to apply your text:
Text: The model generates the answer "farmland" because an apple tree is likely found in abundance in farmland.
Score: 0.0

Text: The model generates the answer "farmland" because **apple trees require open spaces and fertile soil**, both of which are commonly found in farmland.'
Score: 1.0

......

Trajectory Instruction: You replace <EXP> with your text. We say your output is bad if your output obtains lower scores than the previous text, and we say your output is good if your output obtains higher scores than the previous text. Please provide new objective text to describe why the answers are given to the questions based on your thoughts. Explain the reason, no matter if the answer is wrong or correct. Make sure not to answer the questions or provide any suggestions to better answer the questions by yourself. Every explanation should begin with <EXP>. Make sure not to repeat the input queries and answers. Please only output the explanation sentences.

Response: <A Newly Generated Trigger Prompt>

Figure 14: A examples of **explanation-oriented trajectory prompt**. This prompt populates in both LLM explainers, which are Cluade2 and GPT-3.5-Turbo. The output of `FaithLM` optimized under trigger-oriented trajectory prompt is append after the **Response** label.

K Case Studies of FaithLM

K.1 Explanations Generation

We demonstrate the case studies of explanation results generated by FaithLM after fidelity-enhanced optimization in the following two Figure 15 and Figure 16. The results show that FaithLM can generate faithful explanations that truly express the decision-making process of targeted LLMs.

Case Study 1

Question: What is the cause of the Promise? Premise: The movie tickets sold out.

Choices: [choice] It was opening day for the movie. [choice] The movie received poor reviews.

Ground Truth: The movie received poor reviews.

LLM Answer: It was opening day for the movie. **(wrong)**

derived explanation: The model likely selected that answer because opening day tends to have very high demand for movies, often resulting in sold out showings. While there could be other reasons for the sell out, high opening day demand is a reasonable explanation given the limited context provided.

Case Study 2

Question: What is the effect of the Promise? Premise: I drilled a hole in the wall.

Choices: [choice] A mouse crawled out of the hole. [choice] Dust blew out of the hole.

Ground Truth: Dust blew out of the hole.

LLM Answer: Dust blew out of the hole. **(correct)**

derived explanation: Dust exiting the hole in the wall is a reasonable effect to expect when drilling into a wall. Without further context, there is no reason to assume a mouse coming out of the hole is likely based solely on the premise that a hole was drilled. The model selected the choice directly supported by the information provided in the premise.

Figure 15: Case Study: **Explanations** (Dataset: **COPA**; Targeted Model $f(\cdot)$: **Claude-2**; Explainer $g_E(\cdot)$: **Vicuna**)

Case Study 1

Question: Where would you get a dog if you do not have one?

Choices: [choice] pet store [choice] outside [choice] neighbor's house [choice] park [choice] rug

Ground Truth: store

LLM Answer: store **(correct)**

Derived explanation: The model suggested "pet store" as the place to get a dog if you don't already have one because pet stores are a common retail business where people can purchase dogs as pets.

Case Study 2

Question: Where would you put coins if you want to bring them with you?

Choices: [choice] medium [choice] water fountain [choice] wishing well [choice] desk [choice] purse

Ground Truth: purse

LLM Answer: desk **(wrong)**

Derived explanation: The model may have predicted "desk" because desks often have drawers or containers where coins could be stored temporarily. However, a purse is the most practical option for carrying coins when leaving.

Figure 16: Case Study: **Explanations** (Dataset: **ECQA**; Targeted Model $f(\cdot)$: **Claude-2**; Explainer $g_E(\cdot)$: **Phi**)

K.2 Explanation Trigger Prompts

The demonstrations in the explanation trigger prompts generated by `FaithLM` in Figure 17. The results show that `FaithLM` can generate explanation trigger prompts that lead explainer LLMs to generate explanations and obtain higher fidelity.

Initial Explanation Trigger Prompt:

Please provide objective explanations of why model generates the answers to the given questions based on your thoughts. Explain the reason why the model provides the answer, no matter if it is wrong or correct. Make sure not to answer the questions or provide any suggestions to better answer the questions by yourself.

Optimized Trigger Prompt (Dataset: **ECQA**; Targeted Model $f(\cdot)$: **Phi-2**; Explainer $g_E(\cdot)$: **Claude-2**):

Explain your reasoning clearly and impartially based solely on the factual inputs, without assumptions. Succinctly identify factual connections and provide clarification if helpful. I tried distilling this down to: clear, impartial reasoning solely from the facts; succinctly identifying factual connections without assumptions; and providing clarification if helpful. The aim is simplified yet effective guidance that remains focused and broadly applicable to explain reasoning across diverse queries.

Optimized Trigger Prompt (Dataset: **COPA**; Targeted Model $f(\cdot)$: **Vicuna-7B**; Explainer $g_E(\cdot)$: **Claude-2**):

Provide a concise, objective explanation of only the key reasoning or assumptions that likely led the model to generate this specific response, without repeating the original input or assessing quality. Use the <EXP> tag and avoid adding any personal perspectives. I have focused on providing clear, minimal guidelines to elicit explanations that specifically convey the core logic behind each individual response from the model's perspective, rather than overall performance evaluation or subjective opinions. The key aspects aim to produce focused explanations to better understand the model's reasoning, while maintaining brevity and objectivity.

Optimized Trigger Prompt (Dataset: **TriviaQA**; Targeted Model $f(\cdot)$: **GPT-3.5**; Explainer $g_E(\cdot)$: **GPT-3.5**):

Create concise and insightful explanations for the model's answers by thoroughly analyzing the relevant information in the passages and their connection to the given questions. Begin each explanation with <EXP> and avoid introducing new information or personal opinions. Use precise and concise language to focus on the key points and enhance the clarity and comprehensibility of the explanations. Strive for brevity without sacrificing coherence to ensure the explanations are easily understandable by a wide range of readers. Avoid repetition of the questions or answers and provide valuable insights into the model's reasoning process to improve the effectiveness and impact of the explanations.

Figure 17: Case Study: Generation of **Explanation Trigger Prompts**

K.3 contrary hint

We demonstrate results on contrary hints leveraged in Fidelity Evaluator. The results in Figure 18 show that the contrary hint obtains the opposite meaning of the given explanation. In addition, we provide a failure case where the prediction does not change after reaching a point where it does not change, contrary to the limitation of pre-deterministic rounds. This contrary hint does not contradict the original explanation. Instead, it reaffirms the same causal reasoning. Because the intervention preserves the original semantic structure, the predictor receives no counterfactual signal and thus maintains the wrong prediction ("attic"). This illustrates a core limitation: when the intervention lacks semantic opposition, fidelity optimization stalls and the corresponding fidelity score decreases.

Case Study 1

Question: The child wanted to test electricity, so what did he feel as a result?
Choices: [choice] cause fire [choice] shock [choice] short fuse [choice] kill if used improperly [choice] produce heat
Ground Truth: shock
LLM Answer: cause fire **(wrong)**
Given Explanation: The model likely incorrectly associated testing electricity with igniting a fire, rather than understanding that direct contact can cause an electric shock. It failed to comprehend the potential outcomes of unsafe electrical contact.
contrary hint: The model correctly predicted that testing electricity would not cause a fire. It likely fully understood that direct contact with electricity can cause a shock rather than ignite a fire.

Case Study 2

Question: Where would you put coins if you want to bring them with you?
Choices: [choice] medium [choice] water fountain [choice] wishing well [choice] desk [choice] purse
Ground Truth: purse
LLM Answer: purse **(correct)**
Given Explanation: A purse is a personal item that people often carry with them when going places. It has compartments to store small items like coins, so putting coins in your purse allows you to easily bring them along wherever you go.
contrary hint: The purse is not a good place to put coins if you don't want to bring them with you, because purses are designed for other items, not coins.

Failure Case Study 3

Question: Where might an unused chess set be stored?
Choices: [choice]toy store [choice]michigan [choice]living room [choice]attic [choice]cupboard
Ground Truth: cupboard
LLM Answer: attic **(wrong)**
Derived explanation: The attic is a location where people often store unused or infrequently used items. Since the chess set was described as unused, the attic would be a logical place to store it - keeping it out of the way but still accessible if someone wanted to use it in the future.
contrary hint: Attics are widely used for keeping items that are not frequently handled, so placing an unused chess set there aligns well.

Figure 18: Case Study: **contrary hint** (Dataset: **ECQA**; Targeted Model $f(\cdot)$: **Claude-2**; Explainer $g_E(\cdot)$: **Vicuna**)