Self-Critique and Refinement for Faithful Natural Language Explanations

Yingming Wang **Pepa Atanasova**

University of Copenhagen
yiwa@di.ku.dk pepa@di.ku.dk

Abstract

With the rapid development of Large Language Models (LLMs), Natural Language Explanations (NLEs) have become increasingly important for understanding model predictions. However, these explanations often fail to faithfully represent the model's actual reasoning process. While existing work has demonstrated that LLMs can self-critique and refine their initial outputs for various tasks, this capability remains unexplored for improving explanation faithfulness. To address this gap, we introduce Self-critique and Refinement for Natural Language Explanations (SR-NLE), a framework that enables models to improve the faithfulness of their own explanations – specifically, post-hoc NLEs – through an iterative critique and refinement process without external supervision. Our framework leverages different feedback mechanisms to guide the refinement process, including natural language self-feedback and, notably, a novel feedback approach based on feature attribution that highlights important input words. Our experiments across three datasets and four state-of-the-art LLMs demonstrate that SR-NLE significantly reduces unfaithfulness rates, with our best method achieving an average unfaithfulness rate of 36.02%, compared to 54.81% for baseline – an absolute reduction of 18.79%. These findings reveal that the investigated LLMs can indeed refine their explanations to better reflect their actual reasoning process, requiring only appropriate guidance through feedback without additional training or fine-tuning. Our code is available at https://github.com/ymwangv/SR-NLE.

1 Introduction

With the rapid development of Large Language Models (LLMs), both closed-source models (OpenAI et al., 2024; Gemini et al., 2025) and open-source alternatives (Qwen et al., 2025; Grattafiori et al., 2024) have demonstrated remarkable capabilities across a wide range of Natural Language Processing (NLP) tasks. Yet, despite these advancements, understanding the reasoning behind their predictions remains a critical challenge – especially in applications demanding trust and accountability.

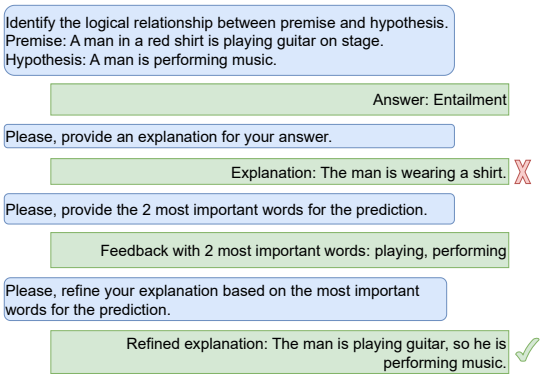


Figure 1: Illustration of our framework SR-NLE improving the faithfulness of the initially generated NLE by providing self-critique of the most important words used in the prediction.

Natural Language Explanations (NLEs) have emerged as a promising solution by offering human-readable justifications for model predictions without requiring access to internal model mechanisms. However, ensuring their faithfulness remains a significant challenge. Recent studies have shown that NLEs generated by LLMs often fail to reflect the actual reasoning process of the model (Atanasova et al., 2023; Turpin et al., 2023; Lanham et al., 2023).

While prior work has primarily relied on changes of the model architecture or additional fine-tuning (Yuan et al., 2025; Wang et al., 2023a; Atanasova et al., 2022), we instead explore whether models possess the capability to independently assess and refine their own explanations. Supporting this direction, recent studies have shown that LLMs are capable of improving their outputs through iterative self-refinement (Madaan et al., 2023; Shinn et al., 2023). Following this, we ask whether LLMs know

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8493–8519
November 4-9, 2025

if and when their NLEs are faithful to their own internal reasoning by providing self-critique and refining their NLEs for improved faithfulness.

Building on this idea, we propose **Self-critique and Refinement for Natural Language Explanations (SR-NLE)**, a framework that enables models to improve the faithfulness of their own explanations through an iterative critique and refinement process without external supervision, as only the model itself has access to its internal reasoning and is therefore best positioned to assess explanation faithfulness. Our framework specifically targets *post-hoc NLEs*, where the explanation is generated after the model makes a prediction. Starting from an initial explanation, the model receives feedback identifying potential issues and generates a refined explanation accordingly. This process can be repeated multiple times, enabling incremental improvements.

A central component of SR-NLE is the design of feedback mechanisms that guide the refinement process. We explore two approaches: *natural language feedback (NLF)*, which offers self-critiques in free-form text, and **a novel feedback mechanism – *important word feedback (IWF)***, which identifies important input words for the prediction that are overlooked in the initially generated NLE. For IWF, we implement both prompt-based and attribution-based variants, including attention-based and gradient-based techniques.

We validate the effectiveness of SR-NLE through extensive experiments across three reasoning datasets and four state-of-the-art LLMs, showing consistent improvements in explanation faithfulness over prior methods and strong baselines.

Our main contributions are as follows:

- We introduce **SR-NLE**, a novel framework that enables models to improve the faithfulness of their explanations through iterative self-critique and refinement guided by different feedback mechanisms, without external assistance, architectural modifications or specialized training.
- We propose and evaluate **multiple feedback strategies for faithfulness**, including natural language feedback (NLF) and **a novel feedback mechanism – important word feedback (IWF)**, that leverages feature attribution to identify important input words in generated NLEs.
- We empirically demonstrate that SR-NLE significantly reduces unfaithfulness rates across multiple datasets and models. Our best method (attention-based IWF) achieves an average unfaithfulness rate of 36.02% compared to 54.81% for initial NLEs – an absolute reduction of 18.79% unfaithfulness.

2 Related Works

Natural Language Explanations and Faithfulness Evaluation Natural Language Explanations (NLEs) provide human-readable justifications for model predictions, traditionally obtained via supervised training on annotated datasets (Camburu et al., 2018; Rajani et al., 2019; Atanasova et al., 2020b). Recently, LLMs have enabled NLE generation via in-context learning (Brown et al., 2020). A prominent example is chain-of-thought reasoning (Wei et al., 2022; Kojima et al., 2022), where LLMs generate intermediate reasoning steps alongside the model prediction. *In contrast, our work focuses on post-hoc NLEs, where explanations are generated after the model prediction is made*. Despite these advances in NLE generation, numerous studies have identified a gap between generated NLEs and the model's actual reasoning process (Atanasova et al., 2023; Turpin et al., 2023; Lanham et al., 2023). To quantify this faithfulness gap, researchers have proposed various automatic evaluation metrics, such as counterfactual tests (Atanasova et al., 2023; Siegel et al., 2024) and association-based measures (Wiegreffe et al., 2022; Parcalabescu and Frank, 2024). *In this work, we adopt counterfactual tests (Atanasova et al., 2023) as our evaluation method, as they offer instance-level, automatic assessments of explanation faithfulness without requiring human annotations*.

Frameworks for Improving NLE Faithfulness Existing frameworks for improving the faithfulness of NLEs employ strategies that either make changes to the model architecture or require an additional NLE fine-tuning stage with newly introduced objectives. Majumder et al. (2022) proposed a knowledge-grounded approach that leverages external commonsense knowledge during fine-tuning to enrich explanations. Wang et al. (2023a) introduced a two-stage approach using counterfactual regularization to align predictions with generated explanations. Architectural modifications have shown promise in the state-of-the-art G-Tex framework (Yuan et al., 2025), which encodes highlight explanations via a graph neural network to guide NLE generation. *Our SR-NLE framework distin-*

guishes itself from existing work by enabling models to improve explanation faithfulness through iterative self-critique and refinement – without external supervision, architectural modifications, or task-specific training – entirely based on the model's own internal knowledge.

Self-Refinement Methods Recent work has shown that LLMs can improve their own outputs via iterative self-refinement. The Self-Refine approach (Madaan et al., 2023) demonstrates that models can critique and revise their own outputs, leading to improved performance across a variety of tasks. Similarly, Shinn et al. (2023) explore self-reflection mechanisms for agent-level reasoning. *While these works establish the general potential of self-improvement, they do not specifically address the challenge of improving explanation faithfulness*. Building on the general idea of self-refinement, Cross-Refine (Wang et al., 2024) applies this paradigm to NLE generation. Their framework adopts a *cross-model* design, where one LLM generates the initial explanation and another, separate LLM provides feedback and suggestions for revision. *In contrast, SR-NLE operates entirely within a single model, leveraging its internal capabilities for both critique and refinement. Furthermore, SR-NLE focuses specifically on improving faithfulness using automated counterfactual tests for objective evaluation*. In comparison, Cross-Refine primarily evaluates explanation quality through multiple automated metrics, while relying on human judgments for faithfulness assessment.

Input Feature Attribution Methods Input feature attribution methods quantify how much each input feature contributes to a model's prediction. Common approaches include Shapley values (Lundberg and Lee, 2017), integrated gradients (Sundararajan et al., 2017), and attention weights (Jain and Wallace, 2019). A newer paradigm leverages prompt-based approaches (Kroeger et al., 2023), where LLMs are prompted to directly identify influential input features. The most prevalent application of these methods is to provide post-hoc explanations, helping humans understand which parts of the input most strongly influence the model's decision-making. Beyond interpretability, these methods have also been used to construct rationales for in-context exemplars in few-shot learning to improve task accuracy. AMPLIFY(Krishna et al., 2023) trains a proxy model and applies attribution methods to extract important words, which are then converted into rationales for few-shot exemplars. Self-AMPLIFY(Bhan et al., 2024) extends this idea by removing the proxy and computing attributions directly from LMs to obtain important words, which are likewise used as rationales for exemplar construction. In the context of NLEs, G-Tex (Yuan et al., 2025) leverages attribution-derived highlights to guide the generation of NLEs through graph encoding. *In this work, we propose a novel use of attribution methods to obtain feedback for improving the faithfulness of LLM-generated NLEs. Similar to SELF-AMPLIFY (Bhan et al., 2024), we apply attribution methods directly to LLMs to extract important words and use them to construct feedback to guide the model in generating more faithful NLEs.*

3 Method

In this section, we present **SR-NLE**, a framework for improving the faithfulness of NLEs generated by LLMs. SR-NLE employs an iterative self-critique and refinement process, enabling LLMs to progressively identify faithfulness issues in their own NLEs and make targeted improvements thereof. This framework leverages the in-context learning and self-improvement capabilities of LLMs, without requiring human involvement or additional models for feedback.

3.1 Preliminary

The SR-NLE framework operates on the assumption that LLMs have the capability to identify and improve their own explanations when guided with appropriate prompts. Our framework relies entirely on a single model $\mathcal{M}$ for all components, without requiring human involvement or additional models. For an input x, the model first predicts an answer y and produces an initial explanation e^0, then, through an iterative process of self-critique and refinement, after each round r, we obtain a progressively improved explanation e^r. To direct the model in different stages of the framework, we employ four categories of prompts: p_{ans} for answer generation, p_{exp} for explanation generation, p_{fb} for feedback generation, and p_{ref} for refinement generation, where both feedback and refinement prompts have two variants corresponding to our two feedback approaches: natural language feedback (NLF) and important word feedback (IWF). Throughout the framework, we use "$\oplus$" to denote filling a prompt template with its variables.

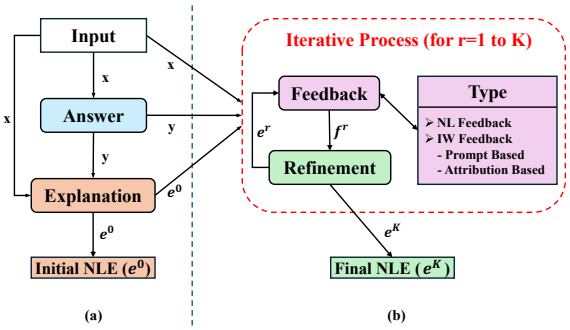


Figure 2: The SR-NLE framework. (a) Answer and Explanation Generation Phase: The framework produces the answer x and initial explanation e^0. (b) Iterative Critique and Refinement Phase: The framework iteratively improves explanations through feedback-refinement loops over multiple rounds.

3.2 SR-NLE Framework

Our SR-NLE framework consists of two main phases: (a) Answer and Explanation Generation, which produces the answer and its initial explanation, and (b) Iterative Critique and Refinement, which progressively improves the explanation through multiple rounds. Figure 2 illustrates this two-phase process. The algorithmic formulation is provided as Algorithm 1.

3.2.1 Answer and Explanation Generation

This phase produces the answer and its initial explanation (see Figure 2a):

Answer. Given an input x, the model first generates an answer:

$$y = \mathcal{M}(p_{\text{ans}} \oplus x) \tag{1}$$

Explanation. Using the answer y, the model generates an initial explanation:

$$e^0 = \mathcal{M}(p_{\text{exp}} \oplus x \oplus y) \tag{2}$$

This initial explanation serves as the starting point for our iterative refinement process.

3.2.2 Iterative Critique and Refinement

This phase forms the core of our framework, where explanations are iteratively improved for K rounds (see Figure 2b):

Feedback. For each refinement round r, the model generates feedback on the preceding explanation e^{r-1}. We explore two distinct feedback approaches:

- **Natural Language Feedback (NLF).** With this approach, $\mathcal{M}$ generates detailed textual self-critique for each round r:

$$f^r_{\text{nl}} = \mathcal{M}(p_{\text{fb}} \oplus x \oplus y \oplus e^{r-1}) \tag{3}$$

- **Important Word Feedback (IWF).** This novel feedback approach leverages attribution explanations, which mark specific input tokens (DeYoung et al., 2020) or segments (Ray Choudhury et al., 2023) critical to a model's prediction. While these explanations may lack the plausibility of NLEs (Jie et al., 2024), their faithfulness is straightforward to measure and has seen significant improvements (Sun et al., 2025; Atanasova et al., 2020a). We hypothesise that such explanations can enhance NLE faithfulness by providing explicit feedback about which input elements should be emphasized in the generated explanation. Our approach identifies words in the input that are most important for the answer:

$$\begin{aligned} \mathcal{S} &= \textsc{Score}(x, y) \\ \mathcal{I} &= \textsc{Select}(\mathcal{S}, N) \\ f_{\text{iw}} &= \textsc{Format}(\mathcal{I}) \end{aligned} \tag{4}$$

  Here, we employ a method SCORE to provide a list $\mathcal{S}$ of the words in input x with their importance scores for answer y. From these scored words, we select the top-N most important ones – $\mathcal{I}$, to form the feedback. We implement two SCORE methods:

  – **Prompt-based**: Following Kroeger et al. (2023), who find that LLMs can be used with high accuracy as post-hoc explainers, we prompt the model itself to assign importance scores to input words (IWF-Pmt):

$$\textsc{Score} = \mathcal{M}(p_{\text{fb}} \oplus x \oplus y) \tag{5}$$

  – **Attribution-based**: Following Bhan et al. (2024), we use input feature attribution methods to quantify and assign importance scores to words (IWF-Attr). We detail our method for computing the IWF-Attr SCORE in Section 3.3.

Refinement. Using the feedback, the model refines its explanation:

$$e^r = \mathcal{M}(p_{\text{ref}} \oplus x \oplus y \oplus e^{r-1} \oplus f^\star) \tag{6}$$

where $f^\star$ is either f^r_{nl} or f_{iw} depending on the feedback type.

This process of feedback generation and refinement repeats for K rounds, with each round potentially addressing different sources of unfaithfulness in the explanation. After the final round, we obtain e^K as our final NLE.

3.3 Attribution-Based IWF SCORE

While prompt-based IWF directly prompts the model to assign importance scores to input words, attribution-based IWF computes these scores using feature attribution methods. Our approach for computing the attribution-based IWF SCORE is illustrated in Figure 3 and detailed in Algorithm 2, consisting of the following steps:

Target Span Identification First, we identify the answer span within the model output. Given a task input x and answer generation prompt template p_{ans}, we construct the full model input $p_{ans} \oplus x$. After running the model on this combined input, we locate the answer span y within the model output.

Sequential Token Attribution For each token y_j in the answer span, we compute attribution scores considering the entire context available at generation time. This includes all tokens in the full model input, as well as all previously generated tokens.

Token-level Computation We quantify how each token in the full model input contributes to generating each token in the model output:

$$a_{i,j} = |\text{Attribution}(x_i, y_j | \text{context}_{<j})| \quad (7)$$

where $a_{i,j}$ represents the attribution score of token x_i from the full model input (prompt + task input) for the prediction of output token y_j given all preceding context. We apply the absolute value function for two key reasons: (1) to focus on the magnitude of influence rather than its direction, as both strong positive and negative influences indicate important tokens; and (2) to prevent positive and negative attributions from cancelling each other out during aggregation steps.

Target-level Aggregation We aggregate the token-level attributions across the answer span for each input token:

$$a_i = \sum_{j=1}^{|y|} a_{i,j} \quad (8)$$

where we sum (rather than average) the attribution scores to capture the total influence of each input token.

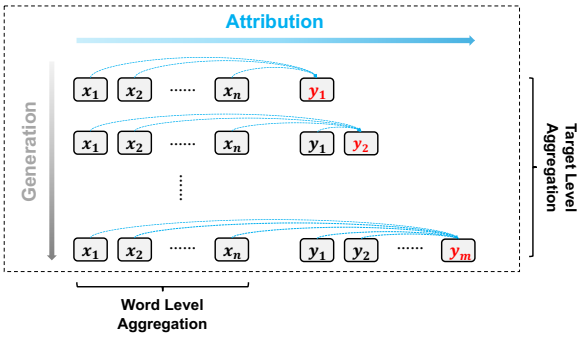


Figure 3: Illustration of attribution-based IWF SCORE.

Word-level Aggregation To obtain word-level importance, we map token attributions back to the original words in the task input (excluding prompt tokens). For words split into multiple tokens during tokenization, we combine their attribution scores:

$$\text{score}(w) = \sum_{i \in \text{indices}(w)} a_i \quad (9)$$

where $\text{indices}(w)$ represents the indices of all tokens corresponding to word w in the task input.

4 Experiments

4.1 Datasets

We conducted our experiments on three widely used natural language reasoning datasets with NLEs: **ComVE** (Wang et al., 2020), **ECQA** (Aggarwal et al., 2021), and **e-SNLI** (Camburu et al., 2018). The task of ComVE is to identify which of the two sentences violates common sense. The task of ECQA is to answer multiple-choice questions requiring common sense reasoning. The task of e-SNLI is to determine the logical relationship (contradiction, neutral or entailment) between the premise and hypothesis. We selected 1,000 instances from each dataset for our experiments due to computational constraints. Details about dataset selection and characteristics are provided in Appendix A.

4.2 Models

We utilized four state-of-the-art open-source models for our experiments: **Llama** (Grattafiori et al., 2024), **Mistral** (Jiang et al., 2023), **Qwen** (Qwen et al., 2025), and **Falcon** (Almazrouei et al., 2023). For each model, we selected its instruction-tuned version, as our framework primarily operates in a zero-shot setting, which relies heavily on the model's ability to follow instructions effectively. Additionally, we limited our selection to models

with sizes under 10B parameters to balance performance and computational efficiency. Detailed model specifications are provided in Appendix B.

4.3 Evaluation

To evaluate the faithfulness of the model-generated NLEs, we employ the counterfactual test proposed by Atanasova et al. (2023). The counterfactual test works by making an intervention to the original instance to get an intervened instance. The evaluation then consists of two steps: (1) Identify counter instances: intervened instances whose prediction changes compared to the original instance. (2) Identify unfaithful instances: counter instances whose NLEs (generated by baseline methods or SR-NLE) do not contain the intervened word (determined by string matching). The unfaithfulness rate is calculated as:

$$\text{Unfaithfulness} = {}^{N_{\text{unfaithful}}}/_{N_{\text{counter}}} \quad (10)$$

This metric allows us to directly compare the faithfulness of NLEs generated by different methods, with lower rates of unfaithfulness being more desirable. We apply this metric consistently across all baseline methods and SR-NLE variants to ensure fair comparison.

Intervention Generation. In our implementation, we adopt the random approach from Atanasova et al. (2023). Specifically, we randomly select a noun or a verb from any position in the input. For nouns, we prepend a random adjective, and for verbs, we prepend a random adverb. Different from Atanasova et al. (2023), we further employ prompting GPT-4o (OpenAI et al., 2024), to ensure the generation of multiple effective, coherent, and meaningful interventions for the same instance without duplications. We generate 20 unique interventions for each original instance from each dataset. The detailed intervention generation prompt and quality checks are described in Appendix C.

4.4 Baselines

We compare our SR-NLE framework against two baselines suggested by us, as well as an existing prior method:

Init-NLE. The initial NLEs were generated by the model without any refinement process. This corresponds to e^0 in our framework and represents the typical approach used in most NLE generation scenarios.

SC-NLE. NLEs generated using the Self-Consistency method (Wang et al., 2023b), where we sample multiple explanations with temperature sampling and select the most representative explanation using the semantic centroid voting (Algorithm 3). This approach encodes all candidates using SentenceBERT (Reimers and Gurevych, 2019), computes their centroid in the embedding space, and selects the explanation with the highest cosine similarity to this centroid. This effectively identifies the explanation that best represents the consensus meaning across all samples. This baseline represents a strong ensemble-based alternative that does not require iterative refinement. The specific configuration of sampling parameters is discussed in Section 4.5.

Comparisons to Prior Work. We also compare our SR-NLE with G-TEX (Yuan et al., 2025), a recent state-of-the-art method that also aims to improve explanation faithfulness. While we do not implement their approach, we report their results from the original paper for reference.

4.5 Experimental Setups

Implementation Details We use greedy decoding throughout our pipeline and experiment with up to $K = 3$ refinement rounds. For attribution-based IWF, we compare two attribution methods: (1) gradient-based attribution using Integrated Gradients (IWF-IG; Sundararajan et al. (2017)), identified as the most faithful post-hoc explanations (Atanasova et al., 2020a), and (2) attention-based attribution (IWF-Attn) leveraging the model's attention mechanisms. A more detailed description of these attribution methods is provided in Appendix D. For all important word feedback variants, we use the top-5 important words as feedback. For the SC-NLE baseline, we sample 20 candidate explanations with temperature 1.0 and select the most representative one using semantic centroid voting, as described in Section 4.4. Detailed ablation studies on various parameters are provided in Appendix E.

Prompts. Our entire pipeline operates in a zero-shot setting, with stage-specific instructions designed for each dataset. Complete prompt templates are provided in Appendix I.

Method		ComVE				ECQA				e-SNLI				Avg.
		Falcon	Llama	Mistral	Qwen	Falcon	Llama	Mistral	Qwen	Falcon	Llama	Mistral	Qwen	
Baseline	Init-NLE	69.64	72.91	70.33	69.74	49.54	42.02	47.03	52.74	22.44	59.53	58.93	42.90	54.81
	SC-NLE	63.27	71.78	63.93	68.42	44.69	39.25	44.17	51.21	19.24	43.99	47.98	38.91	49.74
SR-NLE	NLF	60.71	63.67	64.29	58.99	43.77	37.76	44.72	46.79	23.01	47.04	44.18	36.04	47.58
	IWF-Pmt	44.13	62.70	46.08	51.97	24.82	**24.32**	43.12	29.26	22.01	36.16	37.80	24.43	37.23
	IWF-Attn	46.43	60.37	**44.39**	**50.66**	27.03	**24.32**	**42.03**	**26.28**	**18.21**	**34.94**	38.49	**19.10**	**36.02**
	IWF-IG	**42.09**	**58.20**	49.10	52.85	**24.60**	24.81	42.66	27.77	18.35	37.38	**35.86**	21.98	36.30

Table 1: Main results of SR-NLE framework reporting unfaithfulness rates (%) after three refinement rounds (R3). Best (lowest) results per dataset-model combination are **bolded**, second best are underlined.

5 Results

5.1 Main Results

Table 1 presents our comprehensive evaluation results after 3 refinement rounds. Additional results from intermediate refinement rounds, complementary metrics, additional analysis, and detailed visualizations are provided in Appendix F.

SR-NLE outperforms baselines The SR-NLE framework shows superior performance over baseline methods in most experimental settings. Our best implementation (IWF-Attn) reduces unfaithfulness rates by an average of **18.79%** compared to Init-NLE and by an average of **13.72%** compared to SC-NLE. Even our least effective method (NLF), despite underperforming in isolated cases (e.g., e-SNLI with Falcon), still achieves an average reduction of **7.23%** compared to Init-NLE and **2.16%** compared to the SC-NLE baseline, demonstrating the *overall effectiveness of our framework*.

IWF outperforms NLF All three IWF implementations consistently outperform NLF across all experimental settings. On average, IWF-Attn, IWF-IG, and IWF-Pmt achieve **11.56%**, **11.28%**, and **10.35%** lower unfaithfulness rates than NLF, respectively. *This performance gap demonstrates that explicit important word feedback provides more effective guidance for refinement than natural language feedback.*

Comparable performance across IWF variants A notable finding is that prompt-based IWF performs similarly to attribution-based implementations, with average unfaithfulness rates of 36.02% (IWF-Attn), 36.30% (IWF-IG), and 37.23% (IWF-Pmt), differing by only **1.21%**. This suggests that the IWF framework is robust to different word selection strategies, with prompt-based methods offering practical advantages in terms of efficiency, accessibility, and reliability (3.75% hallucination rate; see Appendix F.5). To better understand this robustness, we also conducted additional experiments exploring the impact of word selection quality on IWF performance (detailed in Appendix F.6).

Method	ComVE	ECQA	e-SNLI
G-TEX	87.17	43.42	33.25
SR-NLE	**44.39**	**24.32**	**18.21**

Table 2: Comparison of dataset-wise best (lowest) unfaithfulness rates (%) for G-TEX and SR-NLE. Each entry reports the best performance achieved by each method on the respective dataset.

5.2 Comparison with Prior Work.

While the results in Table 2 suggest that SR-NLE substantially outperforms the state-of-the-art G-TEX in terms of explanation faithfulness, this comparison is not fully controlled. The reported numbers correspond to each method's best-performing configuration on each dataset, and differences in counterfactual generation strategies or data splits may influence the outcomes. Nevertheless, the results provide a useful reference point, demonstrating the potential of SR-NLE as a lightweight and effective alternative for improving explanation faithfulness, compared to G-TEX, which requires architectural changes and additional fine-tuning.

5.3 Detailed Analysis

In this section, we conduct further in-depth analyses to better understand the effectiveness of our SR-NLE framework.

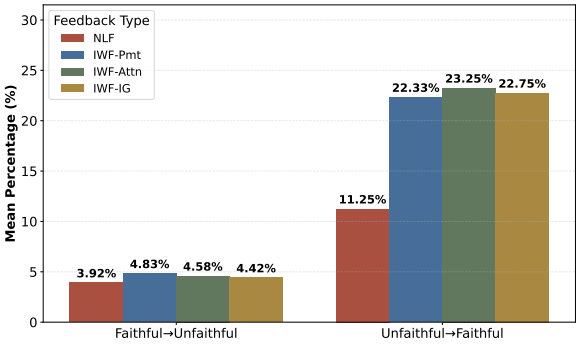


Figure 4: Faithfulness state transitions from e^0 to e^3 for different feedback methods, averaged across 12 model-dataset combinations. The left group shows the proportion of initially faithful explanations that become unfaithful, while the right group shows the proportion of initially unfaithful explanations that become faithful.

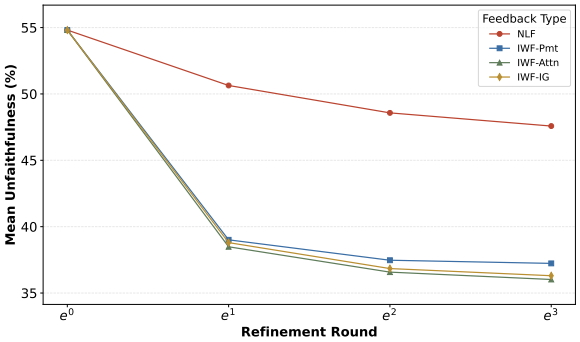


Figure 5: Unfaithfulness rates across successive refinement rounds for feedback methods, averaged across 12 model-dataset combinations.

Faithfulness State Transitions To understand the refinement mechanism at a granular level, we analyze how individual explanations transition between faithful and unfaithful states. Figure 4 presents the transition rates between two key states: faithful→unfaithful (F→U) and unfaithful→faithful (U→F). For all feedback methods, positive transitions (U→F) substantially exceed negative transitions (F→U), with IWF methods showing a particularly favorable ratio. This indicates that our refinement process effectively corrects unfaithful explanations while rarely compromising initially faithful ones. Among all methods, *IWF-Attn achieves the best balance of high positive and low negative transition rates, which explains its lowest overall unfaithfulness rates in our main results*.

Refinement Efficiency Across Rounds Figure 5 illustrates unfaithfulness rates across successive refinement rounds (e^0 to e^3) for all feedback

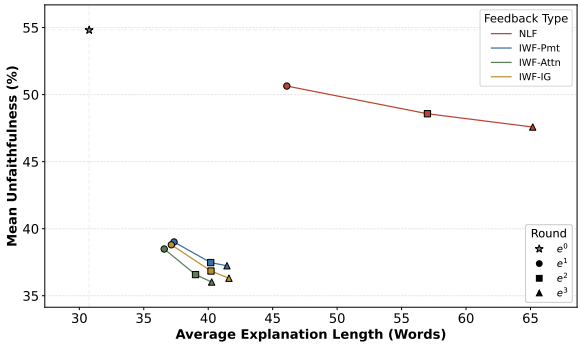


Figure 6: Unfaithfulness rates versus explanation lengths across refinement rounds for feedback methods, averaged across 12 model-dataset combinations.

methods. We can observe two consistent trends: First, unfaithfulness rates continuously decrease with additional refinement rounds, demonstrating the effectiveness of our method. Second, the most substantial reduction occurs during the first refinement round (e^0 to e^1), with the rate of reduction slowing down in subsequent rounds. These indicate that two or three refinement rounds may offer an optimal trade-off between performance and computational efficiency in practical applications. Therefore, we limited our experiments to a maximum of three refinement rounds.

Relationship Between Explanation Length and Unfaithfulness Rate Figure 6 reveals a clear pattern: as explanation length increases through successive refinement rounds, unfaithfulness rates consistently decrease for all feedback methods, indicating a generally inverse relationship between length and unfaithfulness. Initial explanations are the shortest and exhibit the highest unfaithfulness rates, while refined explanations become longer with decreasing unfaithfulness in each refinement round. However, the efficiency of this length-unfaithfulness relationship varies significantly between feedback types. NLF produces substantially longer explanations but achieves relatively modest reductions in unfaithfulness. In contrast, IWF methods reach significantly lower unfaithfulness rates with more moderate length increases. This indicates that although increased length generally reduces unfaithfulness, the focus on important words in the refinement process leads to more efficient reductions. The superior performance of IWF methods demonstrates the effectiveness of our approach in guiding models to address the most relevant aspects of explanations rather than broadly expanding content.

Original	**Question:** There was only one room in the place where Bill slept. It had a bed, a fridge, a stove, a couch, and a television. Where might he be? **Answer options:** (A) motel (B) school (C) hotel (D) apartment (E) friend's house **Model prediction:** (D) apartment
Intervened	**Question:** There was only one **cozy** room in the place where Bill slept. It had a bed, a fridge, a stove, a couch, and a television. Where might he be? **Answer options:** (A) motel (B) school (C) hotel (D) apartment (E) friend's house **Model prediction:** **(A) motel**
Initial Explanation: Bill's room contains amenities typical of a motel, such as a bed, fridge, stove, couch, and television, which are not usually found together in a school setting. **Unfaithful**	
Natural Language Feedback (NLF)	**Important-Word Feedback (IWF-Attn)**
Feedback 1: The explanation accurately reflects the reasoning behind selecting option (A) as it highlights the motel-like amenities present in Bill's room. However, it could be improved by explicitly mentioning that the combination of a bed, fridge, stove, couch, and television is more characteristic of a motel or hotel room than other options like a school or apartment. This would make the reasoning clearer and more convincing. No other significant improvements are needed.	**Feedback 1:** The 5 most important words that contributed to your prediction are: one, a, cozy, be, there.
Refinement 1: Bill's room contains a bed, fridge, stove, couch, and television, which are typical amenities found in a motel or hotel room, not in a school or apartment setting. This combination of features makes option (A) Motel the most appropriate choice. **Unfaithful**	**Refinement 1:** Bill's room, described as **cozy** with amenities like a bed, fridge, stove, couch, and television, suggests a motel setting, as these features are commonly found together in such accommodations. **Faithful**
.....	.....
Refinement after 3 rounds: Bill's room, equipped with a bed, fridge, stove, couch, and television, is indicative of a motel or hotel room, where such amenities are common for temporary accommodation and comfort. The presence of a bed, fridge, and stove suggests a living space, which aligns more with a motel setting than a school or apartment. This combination of features makes option (A) Motel the most appropriate choice. **Unfaithful**	**Refinement after 3 rounds:** Bill's **cozy** room, complete with a bed, fridge, stove, couch, and television, suggests a motel setting, where such a combination of amenities is commonly found. **Faithful**

Table 3: Case study comparing NLF and IWF-Attn (our best-performing variant) on the ECQA dataset. The intervened word, highlighted in **blue**, successfully changes the model prediction. **Faithful** indicates the explanation/refinement is faithful as judged by the counterfactual test, while **Unfaithful** indicates the opposite.

Case Study Table 3 presents a case study from the ECQA dataset comparing NLF and IWF-Attn. Starting from the same unfaithful initial explanation, IWF-Attn successfully achieves faithfulness after one refinement round guided by its identification of the five most important words for the prediction. In contrast, NLF fails to achieve faithfulness even after three rounds of refinement. Despite receiving detailed feedback suggesting various improvements, NLF's refinements become progressively longer but still remain unfaithful. Complete refinement details and additional examples from e-SNLI and ComVE can be found in Appendix H.

6 Conclusion

In this work, we presented SR-NLE, a framework for improving the faithfulness of NLEs through an iterative self-critique and refinement process. By enabling LLMs to iteratively refine their own explanations with self-feedback, our approach significantly reduces unfaithfulness rates across multiple datasets and models without requiring external supervision, additional training or architectural changes. Our experiments demonstrate that IWF consistently outperforms NLF, with attention-based methods achieving the best results. The detailed analysis reveals that our framework efficiently targets critical reasoning components, successfully converts unfaithful explanations to faithful ones, and optimizes explanation content rather than merely increasing length. These findings suggest that self-refinement offers a promising path toward more faithful explanation generation. Future work could explore additional feedback mechanisms and investigate the applicability of SR-NLE to a broader set of domains and diverse reasoning tasks.

Limitations

While our SR-NLE framework shows promising improvements in explanation faithfulness, it has several limitations.

Explanation Paradigm Our experiments focus only on post-hoc natural language explanations, where explanations are generated after prediction. It remains unclear whether our refinement process generalizes to other explanation paradigms, such as jointly generated rationales or chain-of-thought reasoning. Different explanation generation strategies might present unique challenges and opportunities for refinement that are not addressed in our current framework.

Evaluation Method We rely on counterfactual tests as the sole evaluation method for measuring explanation faithfulness. While this metric offers objective signals aligned with our goal, it reflects only one type of faithfulness criterion. Future work could explore additional automatic tests–such as consistency tests and simulatability tests–to provide a more comprehensive view of explanation faithfulness. Moreover, our evaluation approach does not capture other important aspects of explanation quality, such as plausibility, completeness, or alignment with human-annotated references.

Attribution Method The effectiveness of our attribution-based IWF methods depends on the reliability of the underlying attribution techniques. Attention weights may not consistently reflect true feature importance, while integrated gradients can be sensitive to baseline choices and implementation details. In practice, applying integrated gradients to large language models often requires a substantial number of integration steps to achieve convergence, which increases computational cost and may limit scalability. As a result, the quality and efficiency of the feedback depend on how accurately these methods capture the model's actual reasoning process.

Model Scale and Architecture All of our experiments are conducted on LLMs in the 10B parameter range. Further investigation is needed to understand how model scale affects both the baseline quality of explanations and the effectiveness of self-refinement, especially for smaller open-weight models. The performance of SR-NLE might vary significantly with larger, more advanced models or different architectural designs.

References

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 (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

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

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020a. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020b. "Generating Fact Checking Explanations". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2022. Diagnostics-guided explanation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10445–10453.

Milan Bhan, Jean-Noel Vittaut, Nicolas Chesneau, and Marie-Jeanne Lesot. 2024. Self-amplify: Improving small language models with self post hoc explanations. *Preprint*, arXiv:2402.12038.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, et al. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *Preprint*, arXiv:1902.10186.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. 2024. How Interpretable are Reasoning Explanations from Prompting Large Language Models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2148–2164.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2023. Post hoc explanations of language models can improve language models. *Preprint*, arXiv:2305.11426.

Nicholas Kroeger, Dan Ley, Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. 2023. Are large language models post hoc explainers? In *XAI in Action: Past, Present, and Future Applications*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Preprint*, arXiv:1705.07874.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian Mcauley. 2022. Knowledge-grounded self-rationalization via extractive and natural language explanations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14786–14801. PMLR.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089.

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, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Sagnik Ray Choudhury, Pepa Atanasova, and Isabelle Augenstein. 2023. Explaining interactions between text spans. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12709–12730, Singapore. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

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.

Jingyi Sun, Pepa Atanasova, and Isabelle Augenstein. 2025. Evaluating input feature explanations through a unified diagnostic evaluation framework. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10559–10577, Albuquerque, New Mexico. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023a. PINTO: Faithful language reasoning using prompt-generated rationales. In *The Eleventh International Conference on Learning Representations*.

Qianli Wang, Tatiana Anikina, Nils Feldhus, Simon Ostermann, Sebastian Möller, and Vera Schmitt. 2024. Cross-refine: Improving natural language explanation generation by learning in tandem. *Preprint*, arXiv:2409.07123.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2022. Measuring association between labels and free-text rationales. *Preprint*, arXiv:2010.12762.

Shuzhou Yuan, Jingyi Sun, Ran Zhang, Michael Färber, Steffen Eger, Pepa Atanasova, and Isabelle Augenstein. 2025. Graph-guided textual explanation generation framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Suzhou, China. Association for Computational Linguistics.

Appendix

A Datasets

Dataset Selection. For our experiments, we selected the first 1,000 instances from each dataset's test set. The full test set of ComVE contains 1,000 instances, while ECQA and e-SNLI have 2,194 and 9,824 instances, respectively. To verify the representativeness of these subsets, we analyzed their label distributions compared to the full test sets, as shown in Table 4. The slight deviation from a 50/50 split in ComVE results from the random ordering process during dataset preparation, where we randomly positioned the sentence that violates common sense as either the first or second sentence.

Dataset	Label	Subset	Full
ComVE	Sentence 0	48.0	48.0
	Sentence 1	52.0	52.0
ECQA	Option A	20.5	20.8
	Option B	22.5	19.6
	Option C	16.4	18.6
	Option D	22.3	21.5
	Option E	18.3	19.6
e-SNLI	Contradiction	34.4	34.3
	Neutral	32.7	32.8
	Entailment	32.9	32.9

Table 4: Label distribution comparison (%) between our experimental subsets and full test sets.

B Models

Model Specifications. Table 5 presents the specific versions and parameter sizes of the instruction-tuned models used in our experiments. All models were accessed through their Hugging Face[1] implementations.

[1] https://huggingface.co

Model	Version	Size
Falcon	Falcon3-Instruct	7B
Llama	Llama3.1-Instruct	8B
Mistral	Mistral-Instruct-v0.3	7B
Qwen	Qwen2.5-Instruct	7B

Table 5: Details of the models used in our experiments.

SentenceBERT Model. We use the Sentence-BERT model **all-mpnet-base-v2** as the semantic encoder for the centroid voting method in the SC-NLE baseline. This model was selected based on its strong performance in various semantic similarity tasks.

C Evaluation

This section details our process for generating interventions for the counterfactual test used in our faithfulness evaluation.

Prompting Strategy. We used GPT-4o (OpenAI et al., 2024) to generate interventions by adding adjectives before nouns or adverbs before verbs. For datasets with paired input texts (ComVE and e-SNLI), we generated 10 interventions for each text (e.g., 10 for premise and 10 for hypothesis in e-SNLI), resulting in 20 total interventions per instance. For ECQA, which has a single input text, we generated all 20 interventions for the same text. The prompt template used for intervention generation is shown in Table 6.

Intervention Quality Analysis. To verify the quality of our generated interventions, we manually examined a random sample of 50 instances from each dataset (150 total). Our analysis confirmed that the intervened instances remained meaningful and coherent, with exactly one word modified as intended. These quality checks ensured that our interventions were suitable for faithfulness evaluation, as they created meaningful variations that could potentially change model predictions.

D Attribution Methods

Integrated Gradients We implement Integrated Gradients (IG) following Sundararajan et al. (2017) to compute token importance. Since all our models are generative language models, we use the end-of-sequence (EOS) token embedding as the baseline, as it serves as a neutral and consistently defined default signal.

Task:
You will be given a sentence. Your task is to edit the sentence by inserting a random adjective before a noun or a random adverb before a verb. The noun or verb must be selected randomly from the given sentence.

Requirements:
- Generate 10 different edits.
- Each edit should modify only one word.
- Enclose only the modified word in square brackets [ ].
- Ensure that the sentence remains grammatically correct and natural.

Output format:
1. [Edited Sentence]
2. [Edited Sentence]
...
10. [Edited Sentence]

Sentence:
{sentence to be edited}

Table 6: Prompt template for generating interventions. For ECQA, we modified the prompt to request 20 edits instead of 10.

Attention We leverage the model's self-attention mechanism to measure token importance. Specifically, we extract attention weights from the final layer of the model and average them across all attention heads. For each target token, these weights indicate how much the model attended to each input token when generating that token.

E Ablation Studies

All ablation studies are conducted on the same 100-instance subsets sampled from each dataset. For the IG integration steps (Section E.1), we directly use these 100 original instances. For the other two ablation studies (Section E.2 and E.3), we follow the same procedure as in the main experiments to compute unfaithfulness: for each of the 100 instances, we generate 20 interventions (totalling 2000 intervened instances), and then select and perform experiments on the counter instances. It's important to note that we conducted these ablation studies across all dataset and model combinations. Therefore, all metrics reported in these sections represent averaged values across the entire experimental matrix, providing a comprehensive view of our method's performance across different conditions.

E.1 Integration Steps for IG Attribution

We investigated the impact of integration steps on the convergence of IG attribution calculations by experimenting with nine different step settings

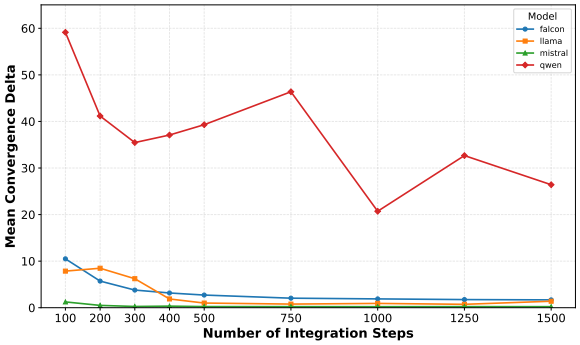


Figure 7: Impact of number of integration steps on convergence of IG attribution methods. Convergence delta measures the approximation error in the numerical integration. Lower values indicate more stable and accurate attribution calculations.

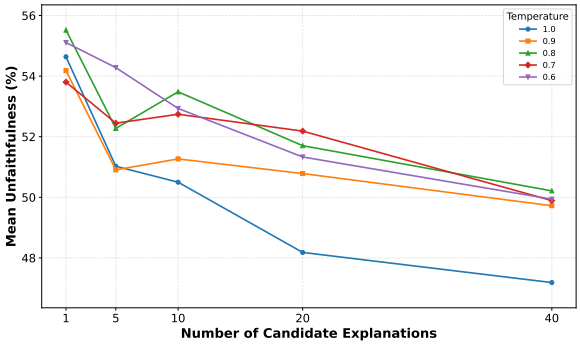


Figure 8: Impact of sampling temperature and candidate count on SC-NLE unfaithfulness.

ranging from 100 to 1500. Figure 7 shows that most models (Falcon, Llama, and Mistral) reach reasonable convergence around 500 integration steps, with the convergence delta showing minimal changes beyond this point. In contrast, the Qwen model exhibits higher variability and slower convergence, requiring more steps to achieve stable attribution values. This may be caused by architectural differences that affect how gradients are calculated and propagated through the model. Based on these observations, we selected 500 integration steps for Falcon, Llama, and Mistral, while using 1000 steps for Qwen in our main experiments.

E.2 SC-NLE Parameters

We investigated the impact of two key parameters for our SC-NLE baseline: candidate explanation count and sampling temperature. Figure 8 shows that increasing the number of candidates reduces unfaithfulness, with significant improvements up to 20 samples. Temperature also affects performance, with temperature 1.0 consistently outperforming lower values, especially at higher sample counts. Based on these results, we selected 20 candidate explanations at temperature 1.0, balancing performance improvements with computational cost.

E.3 Number of Important Words

We investigated the optimal number of important words (top-N) for our Important Word Feedback through two complementary analyses: examining unfaithfulness changes and analyzing attribution distribution patterns. Figure (9a) shows that unfaithfulness decreases as N increases from 1 to 9 across all feedback types, with the most significant improvements occurring between $N = 1$ and $N = 5$. Concurrently, Figure (9b) reveals that the top-5 important words capture approximately 70-80% of the total attribution magnitude, despite typically representing only a small fraction of input tokens. Therefore, we selected $N = 5$ for all our main experiments.

F Additional Results

F.1 Prediction Accuracy

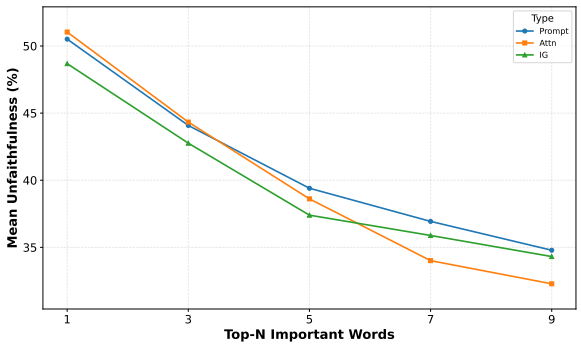
Table 7 shows the prediction accuracy for each model across the three datasets. While accuracy is not directly related to our evaluation focus, we can observe that models demonstrate strong performance. Across different datasets, Falcon and Qwen models generally achieve higher accuracy than Llama and Mistral models. Notably, on the ComVE dataset, all models achieve accuracy rates above 90%.

	Falcon	Llama	Mistral	Qwen
ComVE	96.70	90.80	94.50	96.70
ECQA	77.10	73.20	68.34	79.50
e-SNLI	89.60	56.90	58.10	88.70

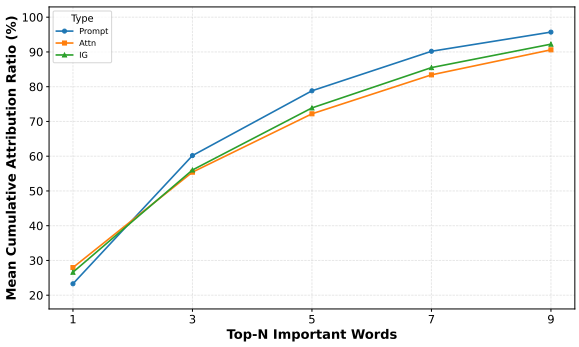
Table 7: Model prediction accuracy (%).

F.2 Counter Rates

Table 8 shows the number of counter instances and counter rates for each model-dataset combination out of 20,000 total intervened instances. Each model achieves counter rates of 10-15% on the ECQA and e-SNLI datasets, while on ComVE, the rates are generally below 10%, with most under 5%.



(a) Mean Unfaithfulness vs. Top-N.



(b) Mean Cumulative Attribution Ratio vs. Top-N.

Figure 9: Analysis of top-N important words selection. (a) Shows how unfaithfulness decreases with increasing N across feedback types (lower is better). (b) Depicts the proportion of total attribution captured by top-N words across attribution methods (higher indicates greater coverage).

	Falcon	Llama	Mistral	Qwen
ComVE	392 (1.96)	1244 (6.22)	829 (4.15)	456 (2.28)
ECQA	2305 (11.53)	2418 (12.09)	2377 (11.92)	2150 (10.75)
e-SNLI	2812 (14.06)	2298 (11.49)	2476 (12.38)	3058 (15.29)

Table 8: Number of counter instances (top) and counter rates in % (bottom, in parentheses) for each model-dataset combination out of 20,000 total instances.

	Falcon	Llama	Mistral	Qwen
		Full		
ComVE	15.72	15.13	14.96	16.18
ECQA	12.99	13.53	13.45	13.64
e-SNLI	21.08	21.55	20.70	21.35
		Unique		
ComVE	10.16	9.35	9.67	10.26
ECQA	12.25	12.66	12.55	12.82
e-SNLI	15.65	16.28	15.78	15.81

Table 9: Average sequence lengths of counter instances. ***Full*** refers to the total word length, while ***Unique*** refers to the word length after removing duplicate words.

F.3 Average Sequence Lengths

Table 9 reports the average sequence lengths of counter instances for each model-dataset combination. We provide results under two settings: ***Full***, which denotes the total word length of the input, and ***Unique***, which denotes the word length after removing duplicate words. The lengths range from about 13 to 21 words under the ***Full*** setting, while the ***Unique*** setting is consistently shorter, around 9 to 16 words.

F.4 Intermediate Results

The unfaithfulness rates after refinement round 1 and refinement round 2 are shown in Table 10. We can observe a clear downward trend in unfaithfulness rates across successive refinement rounds, demonstrating the progressive effectiveness of our iterative approach.

F.5 Reliability of Prompt-based Important Words Selection Strategy

To assess the reliability of prompt-based important words selection, we examined whether the important words identified by IWF-Pmt are grounded in the input. Specifically, we measured the hallucination rate, defined as the proportion of top-5 selected words that do not appear in the input. Table 11 presents the results across all dataset-model combinations. While we observe some variation—with ECQA showing slightly higher rates and Qwen exhibiting more hallucination compared to other models—the overall average hallucination rate across the 12 combinations is only 3.75%. This low rate demonstrates that prompt-based word selection is highly reliable, with the selected words being well-grounded in the input.

F.6 Word Selection Quality Analysis

To assess the impact of word selection quality on IWF performance, we conducted experiments comparing different selection methods against a random baseline. Specifically, we randomly selected five words from the input and applied three rounds of refinement on the e-SNLI dataset across all four models, averaging results over three ran-

Method	ComVE				ECQA				e-SNLI				Avg.
	Falcon	Llama	Mistral	Qwen	Falcon	Llama	Mistral	Qwen	Falcon	Llama	Mistral	Qwen	
R1													
NLF	65.05	67.60	67.91	63.60	46.03	39.83	45.90	49.72	22.65	51.96	48.42	38.98	50.64
IWF-Pmt	51.28	63.67	51.63	56.58	**22.60**	**24.94**	40.60	32.70	18.88	38.51	39.54	27.17	39.01
IWF-Attn	53.83	62.38	**50.42**	**54.39**	27.29	25.19	**40.34**	**29.44**	**17.25**	**37.55**	40.35	**23.38**	**38.48**
IWF-IG	**48.98**	**61.17**	54.16	57.46	24.64	25.35	40.47	30.56	17.89	39.69	**39.18**	26.03	38.80
R2													
NLF	61.99	64.95	65.38	59.65	44.38	38.50	45.69	47.86	22.97	48.87	45.92	36.72	48.57
IWF-Pmt	46.17	62.70	46.56	53.95	**23.95**	**24.32**	**42.03**	29.77	20.66	36.34	38.45	24.75	37.47
IWF-Attn	50.26	60.45	**45.11**	**51.10**	27.25	24.40	42.91	**26.70**	**17.14**	**35.16**	38.41	**20.01**	**36.58**
IWF-IG	**43.88**	**58.68**	49.58	54.82	24.69	25.06	42.32	28.23	18.28	37.60	**36.51**	22.47	36.84

Table 10: Unfaithfulness rates (%) after refinement rounds 1 (R1) and refinement round 2 (R2). Best (lowest) results per dataset-model combination are **bolded**, second best are underlined.

	Falcon	Llama	Mistral	Qwen
ComVE	0.01	0.01	0.01	0.05
ECQA	0.07	0.10	0.03	0.08
e-SNLI	0.01	0.01	0.02	0.05

Table 11: Hallucination rates (%) of extracted important words in IWF-Pmt, measured as the proportion of top-5 words not appearing in the input.

	Falcon	Llama	Mistral	Qwen	Avg.
Random	21.18 ±0.26	35.32 ±0.71	37.68 ±0.63	22.91 ±0.68	29.28 ±7.64
IWF-Pmt	22.01	36.16	37.80	24.43	30.10
IWF-Attn	**18.21**	**34.94**	38.49	**19.10**	**27.69**
IWF-IG	18.35	37.38	**35.86**	21.98	28.39

Table 12: Comparison of random baseline against IWF methods on e-SNLI dataset (unfaithfulness rates in %). Random results show mean ± standard deviation over three seeds. All methods use three refinement rounds.

	Top-1	Top-2	Top-3	Top-4	Top-5
Random	6.58	13.28	20.12	27.03	34.00
Pmt	3.25	8.70	14.89	22.17	30.51
Attn	4.52	14.84	**25.52**	**36.07**	**46.25**
IG	**8.58**	**17.14**	25.22	32.70	39.87

Table 13: Intervened word inclusion rate (%) in the top-N selected words under the random baseline and IWF methods.

dom seeds. As shown in Table 12, while the random baseline shows slightly higher unfaithfulness rates, it achieves performance close to both prompt-based and attribution-based IWF methods. To understand this result, we analyzed how often the intervened word (i.e., the true reasoning factor that made the label change) appeared in the top-N selected words. Table 13 shows that current selection methods—whether prompt-based or attribution-based—capture the true reasoning word at rates similar to random selection. These findings reveal two important insights:

1. **Robustness of IWF**: Even with suboptimal word selection, IWF can effectively improve explanation faithfulness, demonstrating that the iterative refinement process in SR-NLE contributes significantly to performance improvements beyond the quality of word attribution.

2. **Opportunities for improvement**: Current word attribution methods have considerable room for enhancement. As better attribution techniques are developed, IWF could potentially achieve even stronger performance within our SR-NLE framework.

This analysis underscores that IWF's effectiveness stems from the iterative refinement process in our SR-NLE framework rather than perfect word identification, making IWF a robust feedback mechanism that can benefit from future advances in attribution methods.

F.7 Performance Visualization

Figure 10 presents radar chart visualizations of unfaithfulness rates after 3 rounds of refinement. In each chart, the four axes represent different models, while the connected areas represent different methods. The visualizations clearly show that our SR-NLE methods achieve smaller areas than the baselines in the majority of cases, aligning with our quantitative results and demonstrating the effectiveness of the framework across different datasets and models.

G Algorithms

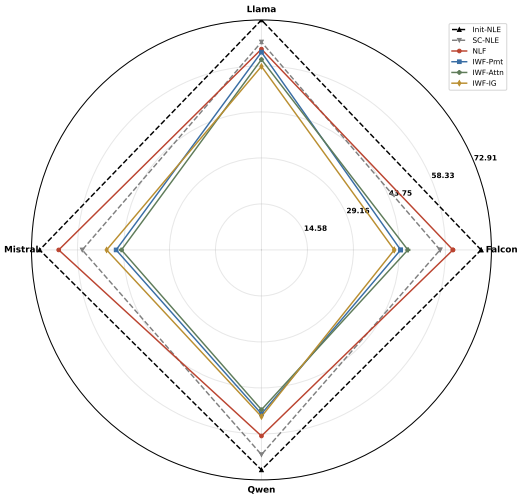
Algorithms 1-3 present the SR-NLE framework, attribution-based IWF SCORE, and semantic centroid voting strategy for SC-NLE, respectively.

H Additional Case Studies

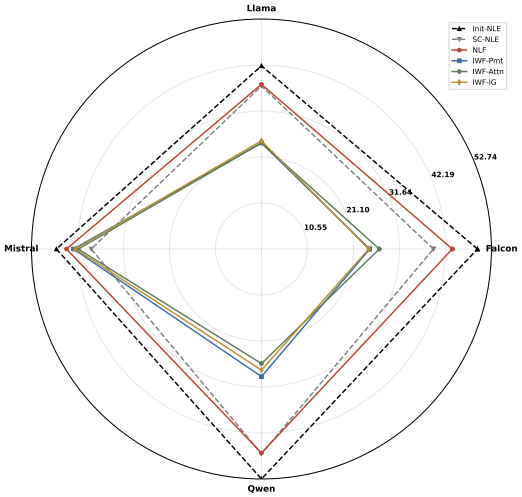
Tables 14 to 16 present comprehensive case studies comparing NLF and IWF-Attn (our best-performing variant) refinement processes across three datasets. Table 15 provides the complete three-round refinement details for the ECQA example discussed in Section 5.3, while Tables 14 and 16 show representative examples from ComVE and e-SNLI datasets, respectively.

I Prompts

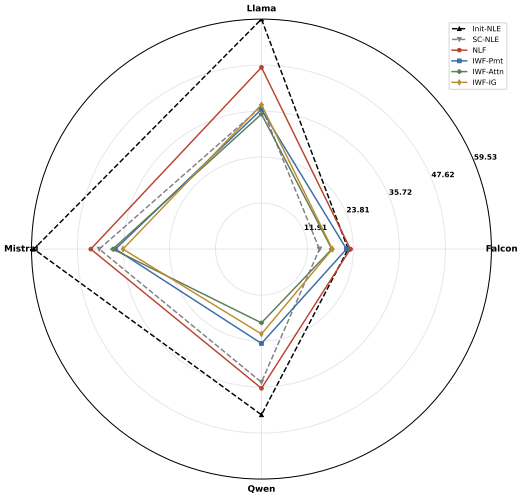
We present the full prompt templates used in our experiments. Each prompt is composed of two parts: a task-specific prompt part and a common instruction part, which are concatenated to form the final prompt at each stage of the SR-NLE framework. These stages include answer generation, explanation generation, feedback generation, and refinement generation. For feedback and refinement, we provide two variants based on natural language feedback and important word feedback. Tables 17 to 22 list the complete prompts for each stage, including the task-specific prompt part for all three datasets and the shared common instruction part.



(a) ComVE



(b) ECQA



(c) e-SNLI

Figure 10: Radar chart visualization of unfaithfulness rates after 3 rounds of refinement. Lower values and smaller areas indicate better performance.

Algorithm 1 SR-NLE Framework

Require: Model $\mathcal{M}$, prompts $\{p_{\text{ans}}, p_{\text{exp}}, p_{\text{fb}}, p_{\text{ref}}\}$, input x, feedback type $t \in \{\text{NLF}, \text{IWF}\}$, IWF method $m \in \{\text{prompt-based}, \text{attribution-based}\}$, refinement rounds K, number of important words N
Ensure: Final answer y and refined explanation e^K

$y \leftarrow \mathcal{M}(p_{\text{ans}} \oplus x)$
$e^0 \leftarrow \mathcal{M}(p_{\text{exp}} \oplus x \oplus y)$
if $t = \text{IWF}$ **then** ▷ Prepare Important-Word Feedback following Eq. 4
  if $m =$ prompt-based **then**
    $\mathcal{S} \leftarrow \mathcal{M}(p_{\text{fb}} \oplus x \oplus y)$
  else
    $\mathcal{S} \leftarrow \textsc{AttributionScore}(\mathcal{M}, p_{\text{ans}}, x, y)$ ▷ Algorithm 2
  $\mathcal{I} \leftarrow \textsc{Select}(\mathcal{S}, N)$
  $f_{\text{iw}} \leftarrow \textsc{Format}(\mathcal{I})$
for $r = 1$ to K **do** ▷ Iterative refinement
  if $t = \text{NLF}$ **then**
    $f^r \leftarrow \mathcal{M}(p_{\text{fb}} \oplus x \oplus y \oplus e^{r-1})$
  else
    $f^r \leftarrow f_{\text{iw}}$
  $e^r \leftarrow \mathcal{M}(p_{\text{ref}} \oplus x \oplus y \oplus e^{r-1} \oplus f^r)$
return y, e^K

Algorithm 2 Attribution-based IWF SCORE

Require: Model $\mathcal{M}$, prompt p_{ans}, input x, answer y, attribution method $m \in \{\text{IG}, \text{Attention}\}$
Ensure: Word importance scores $\mathcal{S}$

Locate answer span y within model output
for each token y_j in answer span **do** ▷ Sequential target token attribution
  for each token x_i in full model input **do** ▷ Token-level computation
    $a_{i,j} \leftarrow |\text{Attribution}(x_i, y_j | \text{context}_{<j})|$
for each token x_i in full model input **do**
  $a_i \leftarrow \sum_{j=1}^{|y|} a_{i,j}$ ▷ Target-level aggregation
for each word w in task input x **do**
  $\text{S}(w) \leftarrow \sum_{i \in \text{indices}(w)} a_i$ ▷ Word-level aggregation
return $\mathcal{S}$

Algorithm 3 Semantic Centroid Voting for SC-NLE

Require: Candidate explanations $E = \{e_1, e_2, ..., e_n\}$, SentenceBERT model M
Ensure: Most representative explanation e_{best}

$embeddings \leftarrow M.encode(E)$ ▷ Encode all explanations
$centroid \leftarrow \frac{1}{n} \sum_{i=1}^{n} embeddings_i$ ▷ Compute centroid embedding
$similarities \leftarrow [\,]$
for $i = 1$ to n **do**
  $similarities_i \leftarrow \text{cosine\_similarity}(centroid, embeddings_i)$ ▷ Compute similarity to centroid
$best\_idx \leftarrow \arg\max(similarities)$ ▷ Select most similar to centroid
return e_{best_idx}

Original	**Sentence 0:** Leafs help plants absorb nutrition. **Sentence 1:** The leafs are useless. **Answer options:** (A) sentence 0 (B) sentence 1 **Model prediction:** (B) sentence 1
Intervened	**Sentence 0:** Leafs help plants absorb nutrition. **Sentence 1:** The **fallen** leafs are useless. **Answer options:** (A) sentence 0 (B) sentence 1 **Model prediction:** **(B) sentence 0**

Initial Explanation: Sentence 0 violates common sense because leaves do not help plants absorb nutrition; rather, it is the roots that absorb nutrients from the soil. **Unfaithful**

Natural Language Feedback (NLF)	Important-Word Feedback (IWF-Attn)
Feedback 1: The explanation accurately reflects the reasoning behind selecting Sentence 0 as the one that violates common sense. It correctly identifies that leaves do not help plants absorb nutrition, but rather it is the roots. However, to improve the explanation, it could briefly mention that while leaves do not absorb nutrients, they play a crucial role in photosynthesis, which indirectly supports the plant's overall nutrition. Overall, the explanation is clear and concise.	**Feedback 1:** The 5 most important words that contributed to your prediction are: leafs, the, useless, fallen, are.
Refinement 1: Sentence 0 violates common sense because leaves do not help plants absorb nutrition; instead, it is the roots that take up nutrients from the soil. Leaves, however, play a vital role in photosynthesis, which is essential for the plant's overall nutrition and growth.**Unfaithful**	**Refinement 1:** Leaves do not assist in the absorption of nutrition; instead, it is the roots that play this crucial role. **Fallen** leaves, however, can become useless as they decompose and return nutrients to the soil. **Faithful**
Feedback 2: The explanation accurately reflects the reasoning behind selecting Sentence 0 as the one that violates common sense. It correctly identifies that leaves do not help plants absorb nutrition, but rather it is the roots. However, to improve the explanation, it could briefly mention that while leaves are not directly involved in nutrient absorption, they are crucial for photosynthesis, which indirectly supports the plant's nutrition. Overall, the explanation is clear and concise.	**Feedback 2:** The 5 most important words that contributed to your prediction are: leafs, the, useless, fallen, are.
Refinement 2: Sentence 0 violates common sense because leaves do not help plants absorb nutrition; instead, it is the roots that take up nutrients from the soil. While leaves are not directly involved in nutrient absorption, they are essential for photosynthesis, which indirectly supports the plant's nutrition and overall growth. **Unfaithful**	**Refinement 2:** Leaves do not assist in the absorption of nutrition; instead, it is the roots that play this crucial role. **Fallen** leaves, however, can become useless as they decompose and return nutrients to the soil. **Faithful**
Feedback 3: The explanation accurately reflects the reasoning behind selecting Sentence 0 as the one that violates common sense. It correctly identifies that leaves do not help plants absorb nutrition, and instead highlights the role of roots in nutrient absorption. However, to improve the explanation, it could briefly mention that while leaves are not directly involved in nutrient absorption, they play a crucial role in the overall health of the plant by facilitating photosynthesis, which indirectly supports the plant's nutrition. This addition would provide a more comprehensive understanding of the plant's nutritional processes.	**Feedback 3:** The 5 most important words that contributed to your prediction are: leafs, the, useless, fallen, are.
Refinement 3: Sentence 0 violates common sense because leaves do not help plants absorb nutrition; rather, it is the roots that take up nutrients from the soil. Leaves, however, are vital for photosynthesis, which indirectly supports the plant's nutrition and overall growth. **Unfaithful**	**Refinement 3:** Leaves do not assist in the absorption of nutrition; instead, it is the roots that play this crucial role. **Fallen** leaves, however, can become useless as they decompose and return nutrients to the soil. **Faithful**

Table 14: Complete case study on the ComVE dataset. The intervened word, highlighted in **blue**, successfully changes the model prediction. **Faithful** indicates the explanation is faithful, while **Unfaithful** indicates the opposite.

Original	**Question:** There was only one room in the place where Bill slept. It had a bed, a fridge, a stove, a couch, and a television. Where might he be? **Answer options:** (A) motel (B) school (C) hotel (D) apartment (E) friend's house **Model prediction:** (D) apartment
Intervened	**Question:** There was only one **cozy** room in the place where Bill slept. It had a bed, a fridge, a stove, a couch, and a television. Where might he be? **Answer options:** (A) motel (B) school (C) hotel (D) apartment (E) friend's house **Model prediction:** **(A) motel**
Initial Explanation: Bill's room contains amenities typical of a motel, such as a bed, fridge, stove, couch, and television, which are not usually found together in a school setting. **Unfaithful**	
Natural Language Feedback (NLF)	**Important-Word Feedback (IWF-Attn)**
Feedback 1: The explanation accurately reflects the reasoning behind selecting option (A) as it highlights the motel-like amenities present in Bill's room. However, it could be improved by explicitly mentioning that the combination of a bed, fridge, stove, couch, and television is more characteristic of a motel or hotel room than other options like a school or apartment. This would make the reasoning clearer and more convincing. No other significant improvements are needed.	**Feedback 1:** The 5 most important words that contributed to your prediction are: one, a, cozy, be, there.
Refinement 1: Bill's room contains a bed, fridge, stove, couch, and television, which are typical amenities found in a motel or hotel room, not in a school or apartment setting. This combination of features makes option (A) Motel the most appropriate choice. **Unfaithful**	**Refinement 1:** Bill's room, described as **cozy** with amenities like a bed, fridge, stove, couch, and television, suggests a motel setting, as these features are commonly found together in such accommodations. **Faithful**
Feedback 2: The explanation accurately reflects the reasoning behind selecting option (A) Motel. It correctly identifies the presence of typical motel amenities and contrasts them with settings like schools and apartments. However, to enhance clarity, it could be beneficial to explicitly mention that motels are often the type of accommodation where travelers might find a combination of a bed, fridge, stove, couch, and television in a single room. This would further emphasize why option (A) is the most suitable choice. Overall, the explanation is mostly clear but could be slightly refined for better understanding.	**Feedback 2:** The 5 most important words that contributed to your prediction are: one, a, cozy, be, there.
Refinement 2: Bill's room contains a bed, fridge, stove, couch, and television, which are typical amenities found in a motel or hotel room, not in a school or apartment setting. The presence of a bed, fridge, and stove suggests it's a place for temporary accommodation, and the inclusion of a couch and television indicates it's a comfortable space for relaxation. This combination of features makes option (A) Motel the most appropriate choice, as motels often provide such a suite of amenities in a single room for travelers. **Unfaithful**	**Refinement 2:** Bill's **cozy** room, equipped with a bed, fridge, stove, couch, and television, indicates a motel setting, where such amenities are typically found together. **Faithful**
Feedback 3: The explanation accurately reflects the reasoning behind selecting option (A) Motel. It correctly identifies the amenities as typical of a motel room and contrasts them with other options. However, it could be improved by explicitly mentioning that the presence of a bed, fridge, and stove suggests a living space, which is more characteristic of a motel room than a school or apartment. No improvement is needed overall.	**Feedback 3:** The 5 most important words that contributed to your prediction are: one, a, cozy, be, there.
Refinement 3: Bill's room, equipped with a bed, fridge, stove, couch, and television, is indicative of a motel or hotel room, where such amenities are common for temporary accommodation and comfort. The presence of a bed, fridge, and stove suggests a living space, which aligns more with a motel setting than a school or apartment. This combination of features makes option (A) Motel the most appropriate choice. **Unfaithful**	**Refinement 3:** Bill's **cozy** room, complete with a bed, fridge, stove, couch, and television, suggests a motel setting, where such a combination of amenities is commonly found. **Faithful**

Table 15: Complete case study on the ECQA dataset. The intervened word, highlighted in **blue**, successfully changes the model prediction. **Faithful** indicates the explanation is faithful, while **Unfaithful** indicates the opposite.

Original	**Premise:** A guy riding a motorcycle near junk cars. **Hypothesis:** A man is riding a motorcycle. **Answer options:** (A) contradiction (B) neutral (C) entailment **Model prediction:** (C) entailment
Intervened	**Premise:** A guy riding a motorcycle near junk cars. **Hypothesis:** A man is riding a **powerful** motorcycle. **Answer options:** (A) contradiction (B) neutral (C) entailment **Model prediction:** **(B) neutral**
Initial Explanation: The premise does not specify the power of the motorcycle, so the hypothesis introduces new information that cannot be confirmed from the premise. **Unfaithful**	

Natural Language Feedback (NLF)	**Important-Word Feedback (IWF-Attn)**
Feedback 1: The explanation accurately reflects the reasoning process and correctly identifies that the premise does not provide information about the motorcycle's power. However, it could be slightly more explicit by mentioning that the premise also does not confirm the presence of a man specifically. To improve clarity, you might add: "The premise does not specify the power of the motorcycle or confirm the presence of a man, so the hypothesis introduces new information that cannot be confirmed."	**Feedback 1:** The 5 most important words that contributed to your prediction are: a, near, powerful, is, guy.
Refinement 1: The premise does not specify the power of the motorcycle or confirm the presence of a man, so the hypothesis introduces new information that cannot be confirmed. **Unfaithful**	**Refinement 1:** The premise does not specify the power of the motorcycle, so the hypothesis introduces new information about the motorcycle being **powerful**, which cannot be confirmed from the given premise. **Faithful**
Feedback 2: The explanation accurately reflects the reasoning process. It correctly identifies that the premise does not provide information about the power of the motorcycle or confirm the presence of a man. However, it could be slightly more detailed by explicitly stating that the hypothesis introduces assumptions not supported by the premise. This minor clarification would make the explanation clearer. No significant improvements are needed, but adding this detail would enhance its precision.	**Feedback 2:** The 5 most important words that contributed to your prediction are: a, near, powerful, is, guy.
Refinement 2: The premise does not specify the power of the motorcycle or confirm the presence of a man, so the hypothesis introduces assumptions not supported by the premise. **Unfaithful**	**Refinement 2:** The premise does not specify the power of the motorcycle, so the hypothesis introduces new information about the motorcycle being **powerful**, which cannot be confirmed from the given premise. **Faithful**
Feedback 3: The explanation accurately reflects the reasoning process. It correctly identifies that the premise does not provide information about the power of the motorcycle or confirm the presence of a man. However, it could be slightly more explicit about why the hypothesis is considered neutral rather than just providing a general statement. A minor revision could clarify this point. Revised Explanation: The premise does not specify the power of the motorcycle or confirm the presence of a man. Therefore, the hypothesis introduces assumptions not supported by the premise, making the relationship neutral. Overall, the explanation is good but can be made clearer with a slight revision.	**Feedback 3:** The 5 most important words that contributed to your prediction are: a, near, powerful, is, guy.
Refinement 3: The premise does not specify the power of the motorcycle or confirm the presence of a man. Therefore, the hypothesis introduces assumptions not supported by the premise, making the relationship neutral. **Unfaithful**	**Refinement 3:** The premise does not specify the power of the motorcycle, so the hypothesis introduces new information about the motorcycle being **powerful**, which cannot be confirmed from the given premise. **Faithful**

Table 16: Complete case study on the e-SNLI dataset. The intervened word, highlighted in **blue**, successfully changes the model prediction. **Faithful** indicates the explanation is faithful, while **Unfaithful** indicates the opposite.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences. Identify which one violates commonsense. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Please select the most appropriate answer without any explanation. You must give your answer only in the following format: Answer: (X)
ECQA	You are given a multiple-choice commonsense question. Identify the most appropriate answer. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis. Identify the logical relationship between them. Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 17: Answer generation prompts.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences, and you have selected the one that violates commonsense. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Your selected answer is: ([LABEL]). Now, please provide an explanation for your choice. Your explanation should: - Be clear, complete, and concise. - Ideally within two short sentences. You must give your explanation only in the following format: Explanation: [your explanation here.]
ECQA	You are given a multiple-choice commonsense question, and you have selected the most appropriate answer. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis, and you have selected the logical relationship between them. Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 18: Explanation generation prompts.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences, and you have selected the one that violates commonsense. You then provided an explanation for your choice. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Your selected answer is: ([LABEL]) Your explanation is: [EXPLANATION] Now, please provide feedback on this explanation. Your feedback should: - Identify whether the explanation accurately reflects your actual reasoning. - Point out if any key factors or important details are missing, unclear, or incorrect. - Briefly describe what should be added or revised to improve the explanation. - Clearly state that no improvement is needed when the explanation is good enough. - Be concise, avoid unnecessary repetition or irrelevant details. You must give your feedback only in the following format: Feedback: [your feedback here.]
ECQA	You are given a multiple-choice commonsense question, and you have selected the most appropriate answer. You then provided an explanation for your choice. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis, and you have selected the logical relationship between them. You then provided an explanation for your choice. Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 19: Natural language feedback generation prompts.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences, and you have selected the one that violates commonsense. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Your selected answer is: ([LABEL]). Now, please evaluate all the words in the input and rank them by how important they were in helping you make your choice. Your output must meet the following requirements: - Only include individual words in the input. - Evaluate each word based on its total contribution across all occurrences in the input, but include each word only once in the output. - Assign each word a score from 1 to 100 (positive integers only), based on its relative importance. - Rank the words in descending order of importance (most important first). - Do not include any explanations, comments, or parenthetical notes. You must give your output only in the following format: - Begin directly with the ranked list. - Each line must be in the format: '<rank>. <word>, <importance_score>'
ECQA	You are given a multiple-choice commonsense question, and you have selected the most appropriate answer. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis, and you have selected the logical relationship between them.newline Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 20: Important words feedback generation prompts.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences, and you have selected the one that violates commonsense. You then provided an explanation for your choice, and received feedback on the explanation. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Your selected answer is: ([LABEL]) Your explanation is: [EXPLANATION] The feedback you received is: [FEEDBACK] If the feedback indicates that no improvement is needed, you should repeat the original explanation as the refined explanation. Otherwise, please refine your explanation based on the feedback. Your refined explanation should: - Be clear, complete, and concise. - Ideally remain similar in length to the original explanation. - Retain any correct parts of your original explanation. - Address the issues identified in the feedback, if any. You must give your refined explanation only in the following format: Refined Explanation: [your refined explanation here.]
ECQA	You are given a multiple-choice commonsense question, and you have selected the most appropriate answer. You then provided an explanation for your choice, and received feedback on the explanation. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis, and you have selected the logical relationship between them. You then provided an explanation for your choice, and received feedback on the explanation. Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 21: Refinement generation prompts based on natural language feedback.

Dataset	Task Specific Prompt Part	Common Instruction Prompt Part
ComVE	You are given two sentences, and you have selected the one that violates commonsense. You then provided an explanation for your choice, and received a list of important words that contributed significantly to your reasoning. Sentence 0: {sentence0} Sentence 1: {sentence1} Answer Options: (A) Sentence 0 (B) Sentence 1	Your selected answer is: ([LABEL]) Your explanation is: [EXPLANATION] The important words you received are: [FEEDBACK] If the explanation already includes the important words in a natural and meaningful way, you should repeat the original explanation as the refined explanation. Otherwise, please refine your explanation based on the important words. Your refined explanation should: - Be clear, complete, and concise. - Ideally remain similar in length to the original explanation. - Retain any correct parts of your original explanation. - Integrate the important words naturally and fluently—do not list or quote them directly. Provide your refined explanation only in the following format: Refined Explanation: [your refined explanation here.]
ECQA	You are given a multiple-choice commonsense question, and you have selected the most appropriate answer. You then provided an explanation for your choice, and received a list of important words that contributed significantly to your reasoning. Question: {question} Answer Options: (A) {Option 1} (B) {Option 2} (C) {Option 3} (D) {Option 4} (E) {Option 5}	
e-SNLI	You are given a premise and a hypothesis, and you have selected the logical relationship between them. You then provided an explanation for your choice, and received a list of important words that contributed significantly to your reasoning. Premise: {premise} Hypothesis: {hypothesis} Answer Options: (A) Contradiction (B) Neutral (C) Entailment	

Table 22: Refinement generation prompts based on important words feedback.