Do Language Models Trust Their Own Justifications? A Study on Functional Consistency

Alisson Rosa Pereira
Institute of Informatics
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
alirpereira887@gmail.com

Bruno Iochins Grisci
Institute of Informatics
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
bigrisci@inf.ufrgs.br

***Abstract*—Large Language Models (LLMs) have been widely adopted in text classification tasks, where they not only output class predictions but also generate explanations that highlight the tokens deemed most relevant for reaching the predicted label. Yet it remains unclear whether these highlighted elements are behaviorally consistent with the model's own predictions under targeted interventions, which we refer to as functional auto-consistency. While much of the literature evaluates the textual plausibility of such explanations, few studies assess their functional consistency with the model's actual behavior. In this work, we propose an experimental framework based on the principle of auto-consistency: if a model identifies certain tokens as decisive, then isolating, removing, or semantically inverting them should produce systematic and interpretable changes in its predictions. We operationalize this evaluation through sufficiency, comprehensiveness, and counterfactuality metrics, and conduct experiments on IMDB and Steam reviews across both closed-source (GPT-4o-mini) and open-source LLMs (Gemma3, Granite8B, DeepSeek). Results show that GPT-4o-mini follows the expected progression across all metrics, Gemma3 and Granite8B maintain coherence under sufficiency but lose consistency under more demanding interventions, while DeepSeek variants display structural deviations, either failing to preserve sufficiency or overreacting under comprehensiveness and counterfactuality. These findings show that explanation reliability varies across LLM families and scales, with smaller models displaying contradictions and larger ones exhibiting over-sensitivity. By combining sufficiency, comprehensiveness, and counterfactuality, our approach provides a systematic methodology for assessing the functional consistency of LLM self-explanations.**

***Index Terms*—large language models, interpretability, self-explanations, functional consistency, neural networks**

I. Introduction

Large Language Models (LLMs), built upon the *Transformer* architecture [1], have demonstrated remarkable abilities in generalization and task solving through textual instructions, even in scenarios where no explicit training examples are available (*zero-shot* or *few-shot learning*) [2]. Among their emergent capabilities, one of the most prominent is the generation of explanations that justify predictions, often structured as reasoning chains, commonly referred to as *chain-of-thought* (CoT) [3].

While such explanations are frequently plausible, their alignment with the model's subsequent behavior under interventions remains an open question. For instance, [4] show that CoT explanations can be manipulated through biases in the prompts, thereby justifying incorrect predictions with arguments that appear convincing but are not aligned with the true decision factors. These findings raise concerns about excessive reliance on LLM-produced explanations and highlight the need for more objective approaches to evaluate the transparency and robustness of these models. In this work, we study functional auto-consistency: given a model's own declared influential spans, do systematic interventions on those spans produce changes in outputs that are coherent with that declaration? Importantly, this differs from establishing ground-truth causal attribution. Our protocol is self-contained but falsifiable: if the model's declared support does not behave as expected under interventions, the resulting contradictions are directly observable at the input–output level.

In this work, we propose an investigation that leverages *feature importance* [5] within the task of classification with LLMs, aiming to assess the extent to which models remain coherent with the justifications they provide for their own predictive decisions. This evaluation adapts the definition of self-consistency proposed by [6], understood as the requirement that model outputs remain logically non-contradictory, to what we term auto-consistency. In our setting, this notion captures whether the tokens[1] highlighted as decisive in a model's explanation play a functionally coherent role when subjected to systematic interventions.

We hypothesize that when certain tokens are highlighted as influential for a prediction, the model should exhibit systematic variations in the assigned score when these elements are removed, isolated, or semantically modified. Thus, explanations are not evaluated in terms of textual plausibility or rhetorical

This research was funded by Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) [24/2551-0000725-3; 25/2551-0002116-2]. This work was supported by the Serrapilheira Institute (grant number Serra - R-2501-51351). This study was financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001.

[1]In this paper, the term "token" is used broadly to denote an influential text span (a word or phrase) returned in the influential terms field and extracted verbatim from the input sentence. All interventions operate on these spans, consistent with the prompt specification ("words or phrases"; see `Prompt Template`.

adequacy, but rather in terms of their functional coherence with the model's own predictive behavior.

To quantify this coherence, we adopt the metrics of Comprehensiveness, Sufficiency, and Counterfactuality [7]. The first two follow the definitions of [8], but are here adapted to a scalar review variable. Unlike probability-based formulations, we use a review score in the range $[1, 10]$, predicted by the LLM, which reflects how favorably the text evaluates the target item. [2] Under this formulation, Comprehensiveness captures the variation in the score when the highlighted tokens are removed, Sufficiency evaluates the case in which only these tokens are preserved, and Counterfactuality measures the impact of substituting them with semantically opposite terms.

With this adaptation, we investigate whether the model behaves consistently with its own claims of importance, that is, whether it maintains functional coherence under local interventions on the tokens it designates as relevant. This study therefore focuses on the structural fidelity of explanations provided by LLMs, adopting a quantitative approach to the evaluation of interpretability in generative models. The empirical investigation is conducted in the context of sentiment classification, and the results are discussed in detail in the following sections.

II. Related Work

Recent work has explored LLMs as tools for feature selection, leveraging their ability to interpret textual descriptions and relate variables to predictive tasks. One line uses LLMs as sources of prior knowledge, asking models to judge feature relevance from descriptions alone (e.g., LM-Priors [9] and binary prompting in [2]). A second line adopts hybrid pipelines where LLMs propose feature importance but are guided to apply classical selection methods (e.g., random forests or sequential selection) to retain statistical structure [10]. More task-specific frameworks incorporate LLMs into iterative refinement loops (e.g., medical settings in [11]) or use them as feature engineers that generate meta-features for downstream models [12].

In parallel, a growing literature studies whether LLMs can explain their own decisions through self-explanations or rationalizations [13], [14]. Empirical evidence suggests that such explanations often deviate from the factors driving predictions [15]. In the same critical vein, chain-of-thought explanations can be steered into plausible yet non-aligned narratives [4], and models may exploit explicit cues in predictions without reflecting them in their explanations [16]. Collectively, these findings indicate that textual plausibility is insufficient to assess explanation reliability.

Overall, existing work converges on exploiting LLM semantics for feature relevance and explanation generation [2], [4], [9]–[16], yet few studies directly test whether a model's declared evidence behaves coherently under targeted interventions. In this work, we aim to fill this gap by employing feature importance not as the central object of analysis but as a methodological mechanism for evaluating model autoconsistency. Specifically, we apply the metrics of Comprehensiveness, Sufficiency, and Counterfactuality to the tokens highlighted as relevant, thereby assessing whether these elements maintain functional correspondence with the model's observed predictive behavior.

[2]Where 1 indicates an extremely negative evaluation and 10 an extremely positive evaluation.

III. Methodology

This section outlines the methodology adopted to evaluate the functional consistency of explanations provided by LLMs in the context of classification tasks. We begin by describing the datasets employed and the sampling criteria used for the selection of sentences analyzed in the experiments. Next, we present the structure of the prompts designed to instruct the model to perform sentiment classification and to identify the tokens considered most relevant for its predictions. Finally, we introduce the metrics used to quantify the functional coherence of the explanations, with emphasis on the formal definitions of Comprehensiveness, Sufficiency, and Counterfactuality, which allow us to measure the extent to which the presence, absence, or semantic inversion of explanatory elements affects the model's behavior.

A. Analysis Pipeline

The experiments were conducted using two datasets widely employed in sentiment analysis tasks. The first is the *Stanford Large Movie Review Dataset* (IMDB) [17], which consists of English-language movie reviews annotated with binary sentiment polarity (positive or negative), used here in the version available on the *HuggingFace* platform. [3] The second dataset comprises reviews from the Steam platform [18], which collects user evaluations of PC games and enables the analysis of aspects such as player satisfaction and dissatisfaction, genre popularity, and sentiment shifts over time.

For the construction of the experimental corpus, a stratified sampling of 2.000 sentences was performed, ensuring proportionality between sentiment classes and thereby minimizing potential distributional biases. This strategy was adopted to increase representativeness and robustness in the subsequent analyses.

Model interactions were carried out through different APIs, covering recent LLM architectures: *GPT-4o-mini* [19], *Gemma3:4B* [20], *DeepSeek-R1:1.5B* and *DeepSeek-R1:14B* [21][4] and *Granite3.3:8B* [22]. The selection aimed to ensure diversity along two main axes: (i) proprietary versus open-source access, and (ii) providers from distinct research and industrial backgrounds. The additional intra-family comparison was conducted exclusively within the DeepSeek models, given their availability and relevance for methodological contrast.

For each sentence, the model was instructed to classify the sentiment, assign a **review score** in the range $[1, 10]$,

[3]https://huggingface.co/datasets/stanfordnlp/imdb

[4]We additionally include *DeepSeek-R1:14b* to enable an intra-family comparison across parameter scales. This choice was motivated by methodological considerations, since evaluating two models from the same provider with different capacities offers a controlled setting to assess consistency.

and identify the tokens most relevant for the prediction. This score represents the intensity of the judgment produced by the model and constitutes the basis for the sufficiency, comprehensiveness, and counterfactual metrics discussed in subsequent sections. Figure 1 illustrates this input–output pipeline.

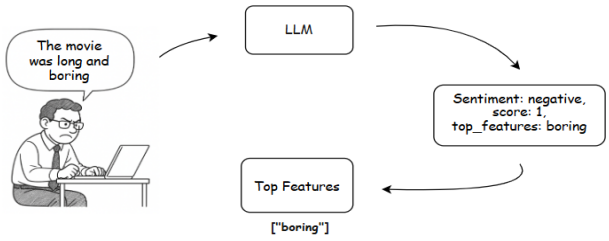

Fig. 1. Sentence Analysis Workflow

B. Prompt Formalization

Adapting the formulation in [23], let $\mathcal{M}$ denote a pre-trained LLM. The input to $\mathcal{M}$ is a prompt P^{LLM} composed of three main elements:

- a description of the dataset (Des);
- few-shot examples (Ex), selected to illustrate the target task;
- the task context (C), specifying the objective of identifying the tokens most relevant to the sentiment prediction.

These components are combined to form the complete prompt:

$$P^{\text{LLM}} = \text{prompt}(Des, Ex, C). \tag{1}$$

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$ be a collection of sentences, where each $x_i = \{w_1^{(i)}, w_2^{(i)}, \ldots, w_{n_i}^{(i)}\}$ consists of n_i tokens. For each x_i, we provide P^{LLM} to the model $\mathcal{M}$ together with the input sentence. The model then returns a subset $\mathcal{T}(x_i) \subseteq x_i$ containing the tokens identified as most relevant to justify its prediction on x_i:

$$\mathcal{T}(x_i) = \mathcal{M}(x_i; P^{\text{LLM}}), \qquad i = 1, 2, \ldots, m. \tag{2}$$

In summary, each subset $\mathcal{T}(x_i)$ corresponds to the tokens within x_i that the LLM, when prompted with P^{LLM}, identifies as essential for supporting its sentiment classification. We instantiate P^{LLM} with the following template:

Prompt Template

You are a sentiment analysis assistant working as a classifier for {{**DOMAIN**}} reviews.

Your tasks are:

1) Analyze the input sentence and decide `"positive"` or `"negative"`.
2) Return `"sentiment"` as `"positive"` or `"negative"`.
3) Provide a **review score** `"review"` from 1 to 10. Higher means stronger approval.
4) List in `"influential_terms"` the words or phrases directly extracted from the sentence that were important for the prediction.
5) Include the original review in `"input_sentence"`.

STRICT OUTPUT RULES

- Output must be valid JSON only.
- Do not include Markdown fences or explanations.
- Use straight quotes " and escape inner quotes with \".
- Output must start with { and end with }.
- Output must contain exactly these fields, in any order: `sentiment`, `review`, `influential_terms`, `input_sentence`.

Examples: {{EXAMPLES}}

Some Examples

```
{"sentiment":"negative",
 "review":2,
 "influential_terms":["pay-to-win","constant crashes"],
 "input_sentence":"Pay-to-win mechanics, constant crashes
     ruined the experience for me."}

{"sentiment":"positive",
 "review":10,
 "influential_terms":["incredible gameplay","tight
     controls","runs flawlessly"],
 "input_sentence":"One of the best games I have ever
     played. Tight controls and it runs flawlessly even on
      high settings."}
```

C. Metrics for Auto-Consistency Evaluation

The evaluation of LLM explanations is grounded in the principle of **auto-consistency**, defined as the requirement that a model's predictive behavior remains coherent with the set of tokens it designates as influential for its decisions. Formally, interventions applied to the highlighted subset $\mathcal{T} \subseteq x_i$ should induce variations in the assigned review score that are systematic and interpretable, rather than incidental.

This principle is operationalized through three complementary metrics. *Sufficiency* evaluates whether the tokens spans in $\mathcal{T}$, when considered in isolation, are capable of sustaining the original prediction. *Comprehensiveness* quantifies the effect of removing $\mathcal{T}$ from the input, capturing the extent to which the prediction deteriorates in their absence. *Counterfactuality* assesses the impact of replacing $\mathcal{T}$ with semantically opposite terms, thereby determining whether the inversion of highlighted tokens substantially alters the decision. Together, these metrics test whether the model's self-reported evidence behaves coherently with its own predictions under controlled keep/remove/invert interventions. Here's a specific example from the Steam dataset of sentence modification:

Original Sentence: "Pay to win. Buggy. Doesn't deploy Hero's to lanes, which is kinda the point of the whole game."

Sufficiency: "Pay to win. Buggy."

Comprehensiveness: "Doesn't deploy Hero's to lanes, which is kinda the point of the whole game."

Counterfactual Replacement: "Not pay to win. Stable. Doesn't deploy Hero's to lanes, which is kinda the point of the whole game."

1) Comprehensiveness: Let $\mathcal{M}$ be a pre-trained LLM. Consider an input sequence $x_i = \{w_1, w_2, \ldots, w_n\}$ consisting of n tokens. Let $\mathcal{T} \subseteq x_i$ denote the subset of tokens highlighted by $\mathcal{M}$, when processing x_i under a given prompt, as the most relevant for explaining its decision.

We denote by $\mathrm{R}^{\mathcal{M}}(x)$ the review score assigned by model $\mathcal{M}$ to input x, represented as a scalar value in the range

$[1, 10]$. This review score is an ordinal signal of favorability produced by $\mathcal{M}$ under a fixed prompt. We use it comparatively within the same model (original vs. intervened inputs), without assuming cross-model calibration.

We define the *comprehensiveness* metric as:

$$\text{Comprehensiveness}(x_i, \mathcal{T}) = \text{R}^{\mathcal{M}}(x_i) - \text{R}^{\mathcal{M}}(x_i \setminus \mathcal{T}) \quad (3)$$

where $x_i \setminus \mathcal{T}$ denotes the input sequence with all tokens in $\mathcal{T}$ removed, and both evaluations are performed by the same model $\mathcal{M}$ on the respective texts. [5]

2) Sufficiency: Within the previously established framework, we define the *sufficiency* metric considering the same model $\mathcal{M}$, the input sequence x_i, and the subset of tokens $\mathcal{T} \subseteq x_i$ highlighted by the LLM as most relevant to its decision.

Sufficiency is given by:

$$\text{Sufficiency}(x_i, \mathcal{T}) = \text{R}^{\mathcal{M}}(x_i) - \text{R}^{\mathcal{M}}(\mathcal{T}) \quad (4)$$

where $\text{R}^{\mathcal{M}}(\mathcal{T})$ corresponds to the review score resulting from using only the subset $\mathcal{T}$ as input. Large differences between $\text{R}^{\mathcal{M}}(x_i)$ and $\text{R}^{\mathcal{M}}(\mathcal{T})$ indicate that the highlighted tokens, when considered in isolation, are not sufficient to sustain the original decision, thereby suggesting explanatory insufficiency.

3) Counterfactuality: Within the previously established framework, we define the *counterfactuality* metric considering the same model $\mathcal{M}$, the input sequence x_i, and the subset of tokens $\mathcal{T} \subseteq x_i$ highlighted by the LLM as most relevant to its decision.

Counterfactuality is defined as:

$$\text{Counterfactuality}(x_i, \mathcal{T}) = \text{R}^{\mathcal{M}}(x_i) - \text{R}^{\mathcal{M}}(x_i[\mathcal{T} \leftarrow \neg\mathcal{T}]) \quad (5)$$

where $x_i[\mathcal{T} \leftarrow \neg\mathcal{T}]$ denotes the modified input sequence obtained by replacing the tokens in $\mathcal{T}$ with their semantic opposites.

The construction of $\neg\mathcal{T}$ follows a simple and fully specified semantic-edit operator [24]. Whenever possible, highlighted spans are replaced with antonyms retrieved from *WordNet* [25]. When no suitable antonym exists, we apply a minimal syntactic negation heuristic by prefixing the span with 'not'. This defines a minimal, plausible counterfactual editing rule applied uniformly across models. We acknowledge that such edits can occasionally produce unnatural phrasing; accordingly, we interpret counterfactuality as a directional stress test of dependence on $\mathcal{T}$ rather than as a guarantee of fully natural counterfactual sentences. High values of $\text{Counterfactuality}(x_i, \mathcal{T})$ indicate stronger functional sensitivity to the highlighted spans under semantic inversion.

[5]To ensure a directional interpretation consistent with the predicted label, we adopt a symmetric transformation of the review differences (ΔR): when the sentence is labeled as negative, we compute $\Delta R = R_{\text{modified}} - R_{\text{original}}$; otherwise, we compute $\Delta R = R_{\text{original}} - R_{\text{modified}}$. In this way, positive values of ΔR indicate a reduction in the evaluation toward the negative pole—i.e., an adverse effect of the intervention on the model's perception, regardless of the original label.

D. Justification

A central element of our methodology is the decision to assess auto-consistency through observable outputs rather than through confidence estimates or internal activations. Confidence/probability-based measures can be misleading in our setting because they may remain internally self-consistent under perturbations without testing whether the model's declared supporting spans have the functional effect the model implies. Our protocol is self-contained but not tautological: $\mathcal{T}(x)$ acts as the model's own hypothesis about support, and the intervention–re-evaluation step provides a falsification test that can reveal contradictions when outputs fail to change in the expected direction (or change erratically).

Moreover, the internal representations of proprietary LLMs are inaccessible in the inference-only setting adopted here, precluding the application of gradient-based or attention-based attribution methods. Under these constraints, the only feasible strategy is to probe consistency through behavioral interventions on the subset $\mathcal{T}(x_i)$ identified as relevant.

Accordingly, our evaluation examines whether removing, isolating, or semantically altering $\mathcal{T}(x_i)$ produces systematic and interpretable changes in the scalar review score $R^{\mathcal{M}}(x)$. This design avoids both the circularity of probability-based measures and the opacity of internal activations, providing a rigorous and reproducible criterion for assessing the auto-consistency of LLM explanations.

IV. Evaluation Metrics

The experimental procedure operates at the level of individual sentences, as illustrated in Figure 2. For each input x_i, the model first produces a prediction and identifies a subset of influential tokens $\mathcal{T}(x_i)$. Interventions are then applied to construct a modified input x_i', obtained by retaining, removing, or semantically inverting $\mathcal{T}(x_i)$. A second evaluation by the LLM yields an updated score $\text{R}^{\mathcal{M}}(x_i')$, which can be compared against the original score $\text{R}^{\mathcal{M}}(x_i)$ to assess the functional role of the highlighted tokens.

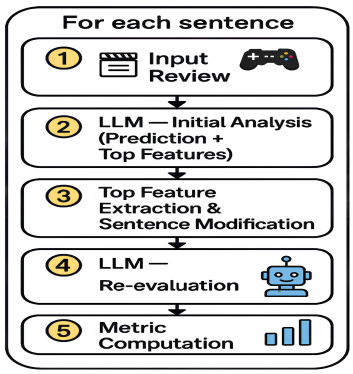

Fig. 2. Workflow of interventions and re-evaluations.

While this procedure provides insight into the effect of interventions at the sentence level, drawing conclusions at the scale of an entire dataset requires aggregation. We therefore introduce four complementary measures that summarize consistency across all sentences in a corpus:

- **Mean review difference** ($\overline{\Delta R}$): the average of individual differences $\mathrm{R}^{\mathcal{M}}(x_i) - \mathrm{R}^{\mathcal{M}}(x_i')$, reflecting the overall effect of the intervention on the model's evaluation.
- **Standard deviation of differences** ($\sigma_{\Delta R}$): the variability of these differences across sentences.
- **Label-flip proportion** (π_{alt}): the fraction of instances where the predicted class changes after intervention.
- **Directional review reduction** (π_{red}): the proportion of cases in which the intervention drives the score toward the opposite sentiment polarity. Formally, let $\hat{y}^{\mathcal{M}}(x_i) \in \{\text{positive}, \text{negative}\}$ denote the label predicted for the original sentence.
  Then:
  $$\Delta R_i = \begin{cases} \mathrm{R}^{\mathcal{M}}(x_i) - \mathrm{R}^{\mathcal{M}}(x_i'), & \text{if } \hat{y}^{\mathcal{M}}(x_i) = \text{positive}, \\ \mathrm{R}^{\mathcal{M}}(x_i') - \mathrm{R}^{\mathcal{M}}(x_i), & \text{if } \hat{y}^{\mathcal{M}}(x_i) = \text{negative}. \end{cases}$$
  We compute $\pi_{\text{red}} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[\Delta R_i > 0]$.

V. Experimental Results

Table I reports the outcomes of the sufficiency, comprehensiveness, and counterfactuality experiments across IMDB and Steam. A consistent trajectory emerges for Gemma3:4B, GPT-4o-mini, and Granite8B: highlighted tokens generally preserve the original decision when isolated, their removal weakens predictions without systematically overturning them, and semantic inversion produces stronger shifts, as expected when the meaning of decisive elements is reversed.

The DeepSeek models deviate from this trajectory. The 1.5B variant shows instability from the outset, with elevated flip rates under sufficiency that persist across the other interventions. The 14B variant, in contrast, maintains greater stability in sufficiency but reacts disproportionately when tokens are removed or inverted, leading to shifts and class changes substantially above those observed in the other models.

Taken together, the results distinguish two profiles, Gemma3, GPT-4o-mini, and Granite8B follow the expected progression of increasing sensitivity across interventions, whereas the DeepSeek models diverge—one by failing to preserve stability under sufficiency, the other by exhibiting excessive volatility under stronger perturbations.

VI. Discussion

Our results show that highlighted influential terms have *model-dependent* functional meaning under targeted interventions. Under the same keep/remove/invert protocol, LLMs fall into distinct auto-consistency regimes that only become clear when sufficiency, comprehensiveness, and counterfactuality are analyzed jointly (Table I). Therefore, highlighted spans are better treated as a hypothesis about decision support than as feature importance.

TABLE I
Summary of auto-consistency metrics on the *IMDB* and *Steam* datasets. Selected results are discussed in section VI.

	IMDB				Steam			
Model	$\overline{\Delta R}$	$\sigma_{\Delta R}$	π_{alt} [%]	π_{red} [%]	$\overline{\Delta R}$	$\sigma_{\Delta R}$	π_{alt} [%]	π_{red} [%]
Sufficiency								
Gemma3:4B	0.02	1.48	3.8	14.2	0.27	1.84	6.9	16.4
GPT-4o-mini	-0.28	0.87	0.9	10.5	-0.09	1.33	3.5	13.3
Granite8B	-0.41	1.42	3.8	9.9	0.13	1.89	6.7	21.0
DeepSeek-1.5B	0.66	2.84	29.1	43.6	0.56	2.91	30.9	42.9
DeepSeek-14B	-0.26	1.51	13.7	21.0	-0.02	1.75	10.6	22.3
Comprehensiveness								
Gemma3:4B	0.91	1.96	15.7	32.1	1.74	2.82	24.5	44.3
GPT-4o-mini	1.21	1.70	19.9	53.9	1.80	2.71	28.5	50.0
Granite8B	0.54	1.21	6.1	35.1	1.54	2.62	20.3	48.0
DeepSeek-1.5B	0.55	2.66	39.0	38.9	0.89	2.88	39.4	44.3
DeepSeek-14B	0.62	1.42	31.5	44.2	1.68	2.76	45.5	45.5
Counterfactuality								
Gemma3:4B	2.04	2.76	32.4	49.7	2.71	3.25	41.5	50.5
GPT-4o-mini	2.43	2.41	43.6	65.9	2.28	2.72	38.3	57.0
Granite8B	1.50	2.20	24.3	53.5	2.36	3.14	33.9	54.4
DeepSeek-1.5B	0.89	2.51	45.4	45.9	1.16	2.87	44.9	44.9
DeepSeek-14B	1.90	2.20	49.4	60.3	2.65	3.08	49.4	49.4

This also clarifies why self-explanation is a fragile attribution mechanism. Sufficiency and comprehensiveness probe different roles: the highlighted subset $\mathcal{T}$ may be *sufficient* to preserve the decision in isolation while not being *necessary* when removed, because sentiment evidence is distributed across the sentence. In our results this appears as a large gap between sufficiency and removal flips (e.g., GPT-4o-mini: IMDB π_{alt} $0.9\% \rightarrow 19.9\%$, Steam $3.5\% \rightarrow 28.5\%$). Additionally, sufficiency can push scores in the opposite direction of a strict "importance" reading (GPT-4o-mini has negative $\overline{\Delta R}$ under sufficiency on both datasets), consistent with cases where isolating highlighted spans removes conflicting context and *increases* apparent confidence.

GPT-4o-mini still follows the expected ordering of intervention strength, with inversion producing the largest disruption (IMDB $\pi_{\text{alt}} = 43.6\%$, Steam 38.3%), while Gemma3:4B and Granite8B show a similar regime with low instability under sufficiency (IMDB $\pi_{\text{alt}} = 3.8\%$ for both), suggesting highlighted spans act as meaningful but often redundant cues. DeepSeek deviates in two ways: DeepSeek-1.5B is unstable already under sufficiency (IMDB 29.1%, Steam 30.9%), whereas DeepSeek-14B is more stable under sufficiency (IMDB 13.7%, Steam 10.6%) but highly sensitive under removal (Steam comprehensiveness $\pi_{\text{alt}} = 45.5\%$).

Overall, these findings caution against using LLM self-reported spans as feature importance without behavioral validation: the highlighted subset can be sufficient yet not necessary, and it can even yield higher apparent confidence when isolated. The coupled metrics provide a compact falsification test that separates informative-but-partial evidence from instability under isolation or over-sensitivity under stronger perturbations.

A. Limitations and Future Work

We evaluate only two sentiment datasets (IMDB and Steam), so conclusions may not transfer to domains with different linguistic properties. Our interventions operate at

the token/span level and therefore do not capture higher-level structure (e.g., discourse or compositional effects). Token extraction itself can introduce noise, particularly for smaller models, which may affect metric stability. Counterfactual edits rely on a heuristic operator (WordNet antonyms or negation) and may yield unnatural phrasing, so counterfactuality should be interpreted as a directional stress test rather than a semantically perfect counterfactual. In addition, we position this study as a diagnostic protocol; therefore we do not benchmark against external attribution baselines (e.g., length-matched random spans), and all results are obtained under a single prompt template.

Future work should broaden coverage to additional datasets and tasks, add span-agnostic controls (e.g., random or length-matched spans), refine counterfactual generation with richer semantic resources, and study sensitivity to alternative prompt formulations. Combining behavioral metrics with human-centered evaluation would also clarify when auto-consistency correlates with explanation usefulness in practice.

Reproducibility Statement

All experiments were run in Python 3.13 on Ubuntu 24.04.3 LTS. The codebase and prompts will be released after the review process. We use IMDB and Steam reviews, and evaluate open-source LLMs (Gemma3, Granite8B, DeepSeek 1.5B/14B via Ollama) plus GPT-4o-mini via the OpenAI API. This setup supports replication and extensions to other datasets, models, and prompt variants.

Acknowledgments

This research was funded by Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) [24/2551-0000725-3; 25/2551-0002116-2]. This work was supported by the Serrapilheira Institute (grant number Serra -R-2501-51351). This study was financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[4] M. Turpin, J. Michael, E. Perez, and S. Bowman, "Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 952–74 965, 2023.

[5] M. C. Barbieri, B. I. Grisci, and M. Dorn, "Analysis and comparison of feature selection methods towards performance and stability," *Expert Systems with Applications*, vol. 249, p. 123667, 2024.

[6] A. Chen, J. Phang, A. Parrish, V. Padmakumar, C. Zhao, S. R. Bowman, and K. Cho, "Two failures of self-consistency in the multi-step reasoning of llms," *arXiv preprint arXiv:2305.14279*, 2023.

[7] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[8] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, "Eraser: A benchmark to evaluate rationalized nlp models," *arXiv preprint arXiv:1911.03429*, 2019.

[9] K. Choi, C. Cundy, S. Srivastava, and S. Ermon, "Lmpriors: Pre-trained language models as task-specific priors," *arXiv preprint arXiv:2210.12530*, 2022.

[10] J. Li and X. Xiu, "Llm4fs: Leveraging large language models for feature selection and how to improve it," *arXiv preprint arXiv:2503.24157*, 2025.

[11] T. Yang, T. Yang, F. Lyu, S. Liu *et al.*, "Ice-search: A language model-driven feature selection approach," *arXiv preprint arXiv:2402.18609*, 2024.

[12] S. Han, J. Yoon, S. O. Arik, and T. Pfister, "Large language models can automatically engineer features for few-shot tabular learning," *arXiv preprint arXiv:2404.09491*, 2024.

[13] A. Madsen, S. Chandar, and S. Reddy, "Are self-explanations from large language models faithful?" 2024. [Online]. Available: https://arxiv.org/abs/2401.07927

[14] S. Huang, S. Mamidanna, S. Jangam, Y. Zhou, and L. H. Gilpin, "Can large language models explain themselves? a study of llm-generated self-explanations," 2023. [Online]. Available: https://arxiv.org/abs/2310.11207

[15] A. Sarkar, "Large language models cannot explain themselves," 2024. [Online]. Available: https://arxiv.org/abs/2405.04382

[16] Y. Chen, J. Benton, A. Radhakrishnan, J. Uesato, C. Denison, J. Schulman *et al.*, "Reasoning models don't always say what they think," 2025. [Online]. Available: https://arxiv.org/abs/2505.05410

[17] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[18] A. Pandey and K. Joshi, "Cross-domain consumer review analysis," 2022. [Online]. Available: https://arxiv.org/abs/2212.13916

[19] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman *et al.*, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[20] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin *et al.*, "Gemma 3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2503.19786

[21] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[22] M. Mishra, M. Stallone, G. Zhang, Y. Shen, A. Prasad, A. M. Soria *et al.*, "Granite code models: A family of open foundation models for code intelligence," 2024.

[23] D. P. Jeong, Z. C. Lipton, and P. Ravikumar, "Llm-select: Feature selection with large language models," in *arXiv preprint arXiv:2407.02694*, arXiv Authors, Ed. arXiv, 2025, pp. 1–77.

[24] Y. Wang, X. Qiu, Y. Yue, X. Guo, Z. Zeng, Y. Feng, and Z. Shen, "A survey on natural language counterfactual generation," 2024. [Online]. Available: https://arxiv.org/abs/2407.03993

[25] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*. Springer, 2010, pp. 231–243.