Faithfulness Evaluation for Decoder-only LLM Attributions with Controlled Retained Information

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are increasingly evaluated with input attribution methods, yet comparing such explanations remains challenging. Existing soft-perturbation faithfulness metrics, such as Soft-NC and Soft-NS, can conflate attribution quality with the number of words retained during perturbation: attribution methods with larger average scores may keep more words and therefore obtain inflated scores. To address this issue, we propose π-Soft-NC and π-Soft-NS, an evaluation framework that compares attribution methods under the same expected retaining probability, thus controlling the number of retained words. We further introduce Grad-ELLM, a gradient-based attribution method tailored to autoregressive decoder-only LLMs, which combines gradient-derived channel importance with attention-derived token importance at each decoding step. Experiments on classification and open-generation tasks with Llama and Mistral show that Grad-ELLM achieves strong comprehensiveness-oriented faithfulness under π-Soft-NC, while there is no dominant method under π-Soft-NS. Our evaluation metric serves as a rigorous framework to compare XAI methods for LLMs, which will support progress in the field.

1 Introduction

Large language models such as Llama have demonstrated their remarkable capabilities across multiple downstream tasks, e.g., text generation and sentiment classification (Grattafiori et al., 2024). However, as these models grow in size and complexity, their "black box" property has raised significant concerns regarding transparency, trustworthiness, and robustness. The lack of interpretability in LLMs hinders their trustworthiness and thus adoption in high-stakes domains, e.g., healthcare (Esteva et al., 2019) and finance (Veale and Binns, 2017), where users need not only accurate predictions but also faithful evidence about the model's decision process. Input attribution methods address this need by assigning importance scores to input tokens (Sundararajan et al., 2017; Lei et al., 2016), where the reliability of the attributions is measured using faithfulness metrics.

Faithfulness is commonly evaluated by perturbing inputs according to attribution scores and measuring the resulting output change (Petsiuk et al., 2018; DeYoung et al., 2020). Hard perturbation metrics, such as deletion, insertion, comprehensiveness, and sufficiency, are intuitive but can create out-of-distribution inputs when tokens are entirely removed or retained (Ancona et al., 2018; Bastings and Filippova, 2020; Yin et al., 2022). Normalized Soft Comprehensiveness (Soft-NC) and Normalized Soft Sufficiency (Soft-NS) (Zhao and Shan, 2024) alleviate this issue by evaluating continuous attribution distributions through soft perturbations.

However, we observe that Soft-NC/NS can yield unfair comparisons when different attribution methods produce score distributions with different means. Since the attribution score directly determines the probability of retaining or removing each token, a method with larger average scores can keep more input information during perturbation. As a result, its faithfulness score may improve simply because the perturbed input contains more information, rather than because the attribution is more faithful. This issue is especially important for decoder-only LLM explanations, where attribution methods can produce highly different score distributions across generation steps.

To address this issue, we propose π-Soft-NC and π-Soft-NS, which evaluate attribution methods under a fixed target retaining probability π. Instead of comparing raw attribution scores directly, we transform each attribution distribution so that different methods retain the same expected amount of information during soft perturbation. This allows faithfulness comparisons to focus on the quality and calibration of the attribution distribution, rather

than on differences in perturbation strength. Moreover, sweeping π reveals how attribution evidence is organized: high π-Soft-NS at small retaining probabilities indicates compact sufficient evidence, while strong π-Soft-NC across retaining probabilities indicates broader comprehensiveness coverage.

In addition to the evaluation framework, we propose Grad-ELLM, a gradient-based attribution method tailored to autoregressive decoder-only LLMs. Grad-ELLM adapts the gradient-attention aggregation idea of Grad-ECLIP (Zhao et al., 2024a) to sequential text generation. Unlike CLIP (Radford et al., 2021), which uses a dual-encoder architecture and a global image-text alignment objective, decoder-only LLMs generate tokens sequentially. Grad-ELLM therefore computes attribution with respect to the next-token logit at each decoding step by combining gradient-derived channel importance with attention-derived token importance, and then aggregates attribution information across generation steps.

In summary, our contributions are four-fold: 1) We identify a bias in existing Soft-NC/NS metrics: attribution methods with different score means can be evaluated under different expected amounts of retained information. 2) We propose π-Soft-NC and π-Soft-NS, faithfulness metrics that control the retained information by comparing attribution methods under the same target retaining probability. 3) We introduce Grad-ELLM, a gradient-based attribution method for autoregressive decoder-only LLMs that combines gradient-derived channel importance with attention-derived token importance. 4) We conduct a comprehensive comparison of representative LLM attribution methods under multiple faithfulness protocols, showing that π-Soft-NC/NS reveal different evidence structures, including compact sufficient evidence, broad comprehensiveness-oriented evidence, and dense but unselective attribution distributions.

2 Related Work

XAI for LLMs includes both local and global explanations (Zhao et al., 2024b; Luo and Specia, 2024); we focus on local input attribution and its faithfulness evaluation for decoder-only LLMs.

2.1 General Input Attribution Methods

Input attribution methods seek to quantify the contribution of input features (e.g., tokens) to model outputs. Early methods were model-agnostic and originated from computer vision or general ML contexts, but have been adapted to NLP. Gradient-based methods, including Saliency, Integrated Gradients, DeepLIFT, and Input$\times$Gradient (Simonyan et al., 2014; Sundararajan et al., 2017; Shrikumar et al., 2017), propagate output signals back to input features and are computationally efficient. However, applying them to decoder-only LLMs is challenging because attribution must be computed across autoregressive generation steps, where noise can accumulate and score distributions can vary substantially across methods (Zhao and Shan, 2024; Fayyaz et al., 2024). Perturbation-based methods, such as LIME (Ribeiro et al., 2016), estimate importance by altering inputs and observing output changes, but text perturbations can break semantic coherence and introduce out-of-distribution artifacts.

Our proposed Grad-ELLM extends gradient-based methods to decoder-only LLMs, incorporating attention mechanisms to handle autoregressive generation without additional forward passes, thus maintaining efficiency while enhancing faithfulness in sequential outputs.

2.2 Attributions for LLMs/Transformers

Recent LLM attribution methods address autoregressive generation and long-context settings. Value Zeroing (Mohebbi et al., 2023) directly measures the effect of zeroing token representations, but requires many forward passes, making it computationally heavy. ReAGent (Zhao and Shan, 2024) uses an auxiliary RoBERTa model (Liu et al., 2019) to generate semantically preserving perturbations, improving perturbation quality at the cost of additional computation. These methods differ not only in computational cost, but also in the scale and density of their attribution scores, which can affect faithfulness evaluation.

Attention-based explanations are efficient because they reuse internal transformer weights (Vaswani et al., 2017; Bahdanau et al., 2015), but raw attention is not always faithful to model decisions (Jain and Wallace, 2019; Serrano and Smith, 2019). Gradient-attention variants such as Grad-SAM (Barkan et al., 2021) improve over raw attention by incorporating gradient signals, but decoder-only generation still requires step-wise attribution.

For VLMs, Grad-ECLIP (Zhao et al., 2024a) explains CLIP (Radford et al., 2021) by combining gradient-derived channel weights with softened at-

tention maps. It visualizes image-text alignments in a dual-encoder vision-language model, but does not directly address the sequential decoding process required for autoregressive LLMs. Grad-ELLM extends this idea from dual-encoder image-text alignment to autoregressive next-token prediction, where attribution must be computed and evaluated across generation steps.

2.3 Evaluation of Attribution Faithfulness

Faithfulness is commonly assessed through perturbation-based metrics, including insertion/deletion curves (Petsiuk et al., 2018), comprehensiveness/sufficiency (DeYoung et al., 2020), and Area Over the Perturbation Curve (AOPC) (Samek et al., 2016). These metrics test whether perturbing important input features causes large changes in model behavior. Zhao and Shan (2024) propose Soft-NC and Soft-NS to handle continuous attributions. However, these metrics may unfairly penalize methods with different information loss profiles during perturbation (see §3.1).

Other evaluation metrics, such as FaithScore (Jing et al., 2024) and Ragas (Es et al., 2024), target factuality or RAG grounding rather than token-level attribution faithfulness. Our π-Soft-NC/NS metrics build on these faithfulness evaluations by introducing a normalization for expected information loss, ensuring fairer comparisons across attribution methods that generate score distributions with different characteristics, particularly in autoregressive LLM contexts. This control also makes the resulting π-curves more interpretable: they reveal whether an attribution method captures compact sufficient evidence or broad comprehensiveness-oriented evidence.

3 Methodology

We first propose our π-Soft-NC/NS faithfulness metrics. We then give a brief overview of decoder-only LLMs, and then propose our corresponding Grad-ELLM attribution method.

3.1 π-Soft-NC and π-Soft-NS

We propose a modification of Soft-NC/NS, denoted as π-Soft-NC/NS, to obtain a fairer and more comprehensive faithfulness metric.

Soft-NC/NS. Zhao and Shan (2024) proposed Soft-NS and Soft-NC to evaluate the faithfulness of the entire attribution distribution. They argue that traditional sufficiency and comprehensiveness metrics are hindered by an out-of-distribution issue (Ancona et al., 2018; Bastings and Filippova, 2020; Yin et al., 2022). Specifically, hard perturbations—which involve entirely removing or retaining tokens—generate discretely corrupted versions of the original input that may fall outside the model's training distribution. As a result, the model's predictions on these altered inputs are unlikely to align with the reasoning processes used for the original full sentences, potentially leading to misleading insights into the model's mechanisms.

To mitigate this issue, rather than create hard perturbations by thresholding or ranking the attribution scores (as in insertion/deletion metrics), soft perturbations are generated by sampling perturbations according to the attribution scores. To create a soft perturbation of the input text, an independent Bernoulli distribution is applied to randomly mask each input embedding token vector $\mathbf{x}_i$, according to its attribution score s_i,

$$\mathbf{x}'_i = e_i \mathbf{x}_i, \quad e_i \sim \text{Bernoulli}(s_i). \tag{1}$$

Here we assume $s_i \in [0, 1]$, which can be viewed as introducing information loss (perturbation) at the embedding level: if $s_i = 0$, the embedding vector $\mathbf{x}_i$ is zeroed out (complete masking); if $s_i = 1$, the embedding is kept (no masking). We denote the attribution scores for the whole text as $\mathbf{s} = \{s_1, \cdots, s_m\}$.

Denote $X' = \{x'_1, \cdots, x'_m\}$ as the soft-perturbation text based on the attribution scores for the next generated token y_t. Let $\mathbf{P}_{X,t} = [p_{1,t}, \cdots, p_{v,t}]$ be the probability distribution of y_t over the whole vocabulary (of size v) when the input of the model is the original text X, and similarly $\mathbf{P}_{X',t} = [p'_{1,t}, \cdots, p'_{v,t}]$ for when the input is the soft-perturbed text X'. The effect of the perturbation on the output distribution is measured using the Hellinger distance:

$$\Delta\mathbf{P}_{X',t} = H(\mathbf{P}_{X,t}, \mathbf{P}_{X',t}) \tag{2}$$

$$= \frac{1}{\sqrt{2}} \Big[\sum_{i=1}^{v} (\sqrt{p_{i,t}} - \sqrt{p'_{i,t}})^2 \Big]^{1/2}, \tag{3}$$

Finally, Soft-NS is defined as the relative effect of keeping important tokens:

$$\text{Soft-NS}(X, x_t, \mathbf{s}) = \frac{\max(0, \Delta\mathbf{P}_{0,t} - \Delta\mathbf{P}_{X',t})}{\Delta\mathbf{P}_{0,t}}, \tag{4}$$

where $\Delta\mathbf{P}_{X',t}$ is the effect of the soft-perturbation X', and $\Delta\mathbf{P}_{0,t}$ is the effect of the zero baseline (all

zero inputs). Likewise, Soft-NC is defined as the relative effect of removing important tokens,

$$\text{Soft-NC}(X, x_t, \mathbf{s}) = \frac{\Delta \mathbf{P}_{\bar{X}',t}}{\Delta \mathbf{P}_{0,t}}, \quad (5)$$

where $\bar{X}'$ is the soft-perturbation that *removes* tokens, i.e., with masks $e_i \sim \text{Bernoulli}(1 - s_i)$, and $\Delta \mathbf{P}_{\bar{X}',t}$ is its corresponding effect. Since Soft-NC is normalized by the zero-baseline distance $\Delta P_{0,t}$, its value can exceed 1 when the perturbed input induces a larger distributional shift than the zero baseline.

Bias of Soft-NC/NS. Soft-NC/NS has a problem that makes it potentially unfair to compare the performance of different attribution methods. Different attribution methods may yield heatmaps with different mean values, and thus the overall probability of keeping any word (or the expected number of kept words) after soft perturbation may also differ. In particular, given the text attribution scores $\mathbf{s}$, the probability of retaining any word is:

$$\hat{\pi} = \frac{1}{m} \sum\nolimits_{i=1}^{m} \mathbb{E}[e_i] = \frac{1}{m} \sum\nolimits_{i=1}^{m} s_i, \quad (6)$$

and the expected number of kept words is $m\hat{\pi}$. It will be unfair to compare the Soft-NC/NS values for two attribution methods A and B with different retaining probabilities, $\hat{\pi}_a$ and $\hat{\pi}_b$, as the expected amount of retained information is different. That is, the difference in Soft-NC/NS scores may be due to different amounts of attributed words, rather than better attribution quality.

π-Soft-NC/NS. To address the above problem, we propose π-Soft-NC/NS, which first transforms the attribution scores so that the calculated retaining probability $\hat{\pi}$ matches a given target probability $\pi \in [0, 1]$. Assuming scores $s_i \in [0, 1)$, we adopt the alpha transformation, $\tilde{s}_i = s_i^{\alpha}$, and set α such that the transformed scores match the target retaining probability, $\frac{1}{m} \sum_{i=1}^{m} s_i^{\alpha} = \pi$. Note that target $\pi = 1$ is obtained by setting $\alpha = 0$, and $\pi = 0$ is obtained with $\alpha \to \infty$. We then define:

$$\begin{aligned} \pi\text{-Soft-NC}(X, x_t, \mathbf{s}, \pi) &= \text{Soft-NC}(X, x_t, \mathbf{s}^{\alpha}), \\ \pi\text{-Soft-NS}(X, x_t, \mathbf{s}, \pi) &= \text{Soft-NS}(X, x_t, \mathbf{s}^{\alpha}), \\ \text{where } \frac{1}{m} \sum\nolimits_{i=1}^{m} s_i^{\alpha} &= \pi. \end{aligned} \quad (7)$$

In practice, α is solved numerically, e.g., using bisection search. For π-Soft-NS, π directly corresponds to the expected fraction of retained input information. For π-Soft-NC, the same transformed attribution scores are first calibrated to target π and then complemented to remove high-attribution information – larger π indicates that a larger expected mass of high-attribution information is removed.

Using the same target π, two attribution methods can be fairly compared using π-Soft-NC/NS, where both soft-perturbed text will now have the same expected number of retained words. Because the soft-perturbation is based on attribution scores, attribution methods that perform well according to π-Soft-NC/NS will produce scores that are well-calibrated probabilities of importance. Furthermore, a comprehensive evaluation is obtained by sweeping the value of $\pi \in [0, 1]$ and plotting π-Soft-NC/NS vs. π curves, and then summarizing using AUC. The illustration of the proposed evaluation protocol is shown in Fig. 1.

Evaluation Protocols. The attention mechanism in Transformers dynamically routes information in each layer based on the embeddings of the previous layers. Thus, we propose two evaluation protocols to consider different aspects of evaluation.

- *Dynamic-Routing*: the transformer attention is dynamically computed based on the perturbed inputs. The measured output change reflects both the causal contribution of token content and the model's dynamic re-routing after perturbation.
- *Fixed-Routing*: the transformer attention is fixed based on the non-perturbed input (i.e., the original forward pass). Here changes in the output distribution are mainly caused by changes to the value/content information carried by the perturbed tokens, rather than by attention reallocation.

In summary, the dynamic-routing protocol evaluates how the attributions identify words that *could* be useful for the model under dynamic conditions, while the fixed-routing protocol is a local evaluation of the attributed words under the fixed context of the original input.

3.2 Preliminary on LLMs

Decoder-only LLMs, such as Llama (Grattafiori et al., 2024) and Mistral (Jiang et al., 2023), generate text autoregressively: given an input sequence of tokens $X = \{x_1, x_2, \ldots, x_m\}$, the model predicts output tokens $Y = \{y_1, y_2, \ldots, y_n\}$ sequentially, where each y_t is conditioned on all the inputs and the previous outputs $\{x_1, \ldots, x_m, y_1, \ldots, y_{t-1}\}$. That is, at step t, the model computes the logit l_t for y_t based on the prefix $\{x_1, \ldots, x_m, y_1, \ldots, y_{t-1}\}$.

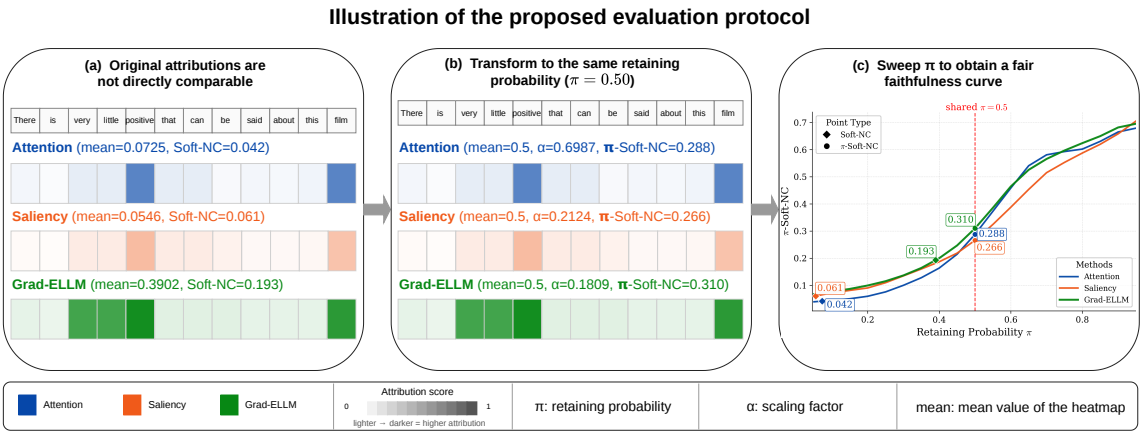


Figure 1: Illustration of the proposed π-Soft evaluation protocol. (a) Different attribution methods can produce score distributions with different means, leading to different expected retaining probabilities and therefore incomparable original Soft-NC values. (b) For a target retaining probability π, each attribution distribution is transformed with its own scaling parameter α so that all methods share the same expected retaining probability; the resulting π-Soft-NC values are therefore comparable. (c) Sweeping π yields the full π-Soft-NC curve. Original Soft-NC values correspond to the special case $\alpha = 1$ and appear as different points on the curve, showing why they are not directly comparable across methods.

The Transformer decoder consists of N layers, each with self-attention and feed-forward networks. In self-attention, for a query $\mathbf{q}$, keys $\mathbf{k}_i$, and values $\mathbf{v}_i$, the attention weights are:

$$\lambda_i = \mathrm{softmax}(\frac{\mathbf{q}\cdot\mathbf{k}_i^\top}{\sqrt{d}}), \tag{8}$$

and the output is $\sum_i \lambda_i \mathbf{v}_i$, where d is the embedding dimension. The attention weight captures token dependencies, but does not necessarily reflect actual contributions to the output (Jain and Wallace, 2019), which necessitates explanation methods.

We denote $\mathbf{z}^{(k)} \in \mathbb{R}^{(m+t)\times d}$ as the input to layer k (output of layer $k+1$), with $\mathbf{z}^{(N)}$ as token embeddings (of inputs X) and $\mathbf{z}^{(0)}$ as the final hidden states (see Fig. 2). The logit l_t for the next token y_t is derived from the t-th generated token's hidden state and $\mathbf{z}_t^{(0)} \in \mathbb{R}^d$ is the corresponding row in $\mathbf{z}^{(0)}$. We denote $\mathbf{o}^{(k)} \in \mathbb{R}^{(m+t)\times d}$ as the output of the self-attention block in layer k, and thus the output of the transformer block k is $\mathbf{z}^{(k-1)} = \mathbf{o}^{(k-1)} + \mathbf{z}^{(k)}$.

3.3 Gradient-based Explanations for LLMs (Grad-ELLM)

In addition to the proposed evaluation metrics, we introduce Grad-ELLM as a gradient-based attribution method for autoregressive decoder-only LLMs. Grad-ELLM explains the logit l_t of the next generated token y_t by combining two sources of information in the self-attention computation: gradient-derived channel importance and attention-derived token importance.

For a transformer layer k, let $\mathbf{o}_t^{(k)}$ be the attention output at the last query position when predicting y_t. The attention output can be written as

$$\mathbf{o}_t^{(k)} = \sum_i \lambda_i^{(k)} \mathbf{v}_i^{(k)}, \tag{9}$$

where $\mathbf{v}_i^{(k)}$ is the value vector of the i-th input token, and $\lambda_i^{(k)}$ is its token weight derived from the query-key similarity. We approximate the contribution of each channel by the gradient of the target logit with respect to the attention output:

$$w_c^{(k)} = \frac{\partial l_t}{\partial \mathbf{o}_t^{(k)}[c]}. \tag{10}$$

The attribution assigned to the i-th token at layer k is then

$$H_i^{(k)} = \mathrm{ReLU}\left(\sum_c w_c^{(k)} \lambda_i^{(k)} \mathbf{v}_{ic}^{(k)}\right). \tag{11}$$

We aggregate $H_i^{(k)}$ over the selected decoder layers and normalize the resulting token scores to obtain the final attribution map. Following Grad-ECLIP (Zhao et al., 2024a), we use normalized query-key similarities as loosened token weights, i.e., $\lambda_i^{(k)} \approx \Phi(\mathbf{q}_t^{(k)} \mathbf{k}_i^{(k)\top})$, where Φ denotes min–max normalization over input tokens. This avoids overly peaky softmax weights and yields a continuous attribution distribution over the input. The complete derivation is provided in Appendix B.

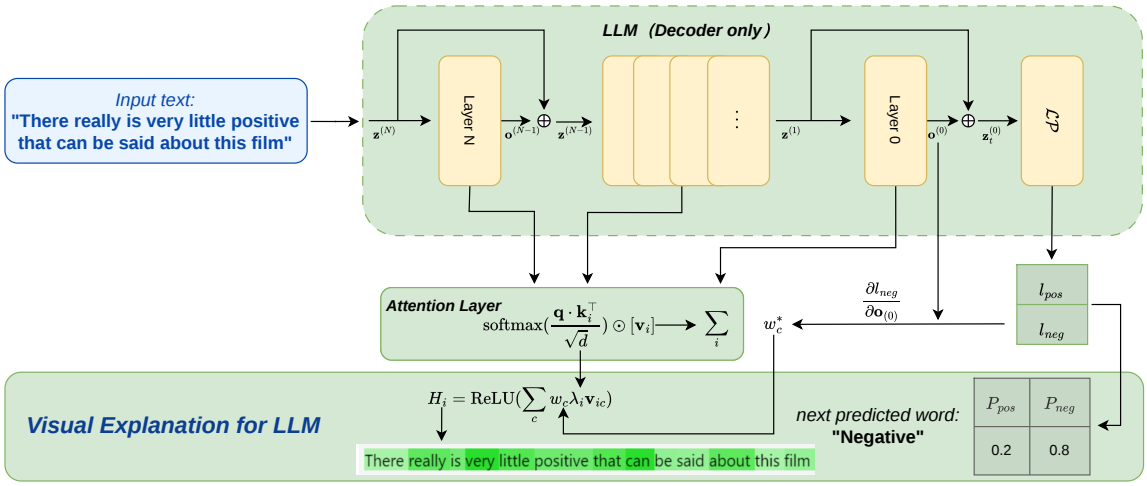


Figure 2: Framework of the proposed Grad-ELLM. An attribution map is generated by aggregating the values $\mathbf{v}$ as feature map in the attention layer with token weight λ_i and channel weight w_c. This example shows the attribution map when generating the word "Negative".

4 Experiments

We evaluate both the proposed π-Soft-NC/NS metrics and Grad-ELLM. We first compare Grad-ELLM with representative attribution baselines under π-Soft-NC/NS, and then analyze complementary ranking-based behavior using insertion/deletion metrics in the Appendix E.

4.1 Setup

We use Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 from Hugging Face Transformers (Wolf et al., 2020). Evaluation on a larger Llama-3.1-70B-Instruct is reported in Appendix G. See Appendix A for implementation details.

We compare with representative baseline XAI methods from each category: 1) attention map-based methods, including raw attention in the last decoder layer (Bahdanau et al., 2015); 2) classical gradient-based methods, such as Saliency (vanilla gradients) (Simonyan et al., 2014), Input×Gradient (Simonyan et al., 2014), Integrated Gradients (IG) (Sundararajan et al., 2017), and DeepLIFT (Shrikumar et al., 2017); 3) perturbation-based Value Zeroing (Mohebbi et al., 2023); 4) a random baseline (Random), which assigns uniform random scores to tokens as a control. 5) Recursive Attribution Generator (ReAGent) (Zhao and Shan, 2024), a model-agnostic feature attribution method that exploits an external RoBERTa model.

4.2 Datasets

We evaluate on three classification datasets and two open-generation datasets following Zhao and Shan (2024). The classification datasets include IMDb (Maas et al., 2011), SST2 (Socher et al., 2013), and BoolQ from ERASER (DeYoung et al., 2020). The open-generation datasets include TellMeWhy (Lal et al., 2021) and WikiBio (Manakul et al., 2023). Detailed dataset descriptions and prompts are provided in Appendix C and D.

4.3 Faithfulness Metrics

We mainly use our proposed π-Soft-NC and π-Soft-NS with *dynamic-routing* protocol to measure the faithfulness of the full importance distribution fairly. We compute curves using $\pi \in \{0.05, 0.10, \ldots, 0.95\}$. For each input and each retaining probability π, π-Soft-NC/NS are estimated by averaging over 15 Monte Carlo samples of the stochastic soft perturbation.

Following Zhao and Shan (2024), for classification tasks with explicit label words, we evaluate the **token-level faithfulness** of the target label word. For open-generation tasks, we evaluate sequence-level faithfulness by computing π-Soft-NC/NS every five generated tokens from the sequential prediction and averaging the resulting scores across evaluation steps.

Due to space limiations, evaluation under the *fixed-routing* protocol is presented in Appendix K.

(a) Llama: AUC of π-Soft-NC/NS (↑)

Dataset	Attn	DL	I×G	IG	Rnd	Sal	VZ	RG	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.603	0.574	0.570	0.585	0.551	0.568	0.592	0.559	0.590
SST2	0.592	0.588	0.588	0.577	0.549	0.577	0.594	0.562	0.586
BoolQ	0.337	0.382	0.381	0.357	0.291	0.384	0.342	0.310	0.345
TellMeWhy	0.473	0.506	0.504	0.526	0.392	0.500	0.474	0.463	0.453
WikiBio	0.458	0.476	0.475	0.476	0.352	0.472	0.471	0.401	0.459
Avg	0.493	0.505	0.504	0.504	0.427	0.500	0.495	0.459	0.487
AUC π-Soft-NC ↑									
IMDb	0.433	0.404	0.402	0.410	0.412	0.399	0.420	0.405	0.437
SST2	0.296	0.306	0.303	0.304	0.312	0.290	0.304	0.305	0.315
BoolQ	1.574	2.228	2.195	2.153	1.888	2.221	1.557	1.901	2.398
TellMeWhy	0.754	0.756	0.749	0.693	0.622	0.749	0.736	0.542	0.649
WikiBio	0.785	0.743	0.740	0.762	0.663	0.732	0.812	0.652	0.797
Avg	0.768	0.887	0.878	0.864	0.779	0.878	0.766	0.761	0.919

(b) Mistral: AUC of π-Soft-NC/NS (↑)

Dataset	Attn	DL	I×G	IG	Rnd	Sal	VZ	RG	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.495	0.482	0.484	0.463	0.401	0.479	0.526	0.425	0.481
SST2	0.453	0.437	0.439	0.451	0.317	0.443	0.474	0.324	0.414
BoolQ	0.360	0.356	0.357	0.352	0.312	0.356	0.367	0.407	0.367
TellMeWhy	0.479	0.483	0.483	0.485	0.384	0.486	0.491	0.463	0.465
WikiBio	0.520	0.540	0.570	0.537	0.412	0.573	0.456	0.476	0.541
Avg	0.461	0.460	0.467	0.458	0.365	0.467	0.463	0.419	0.454
AUC π-Soft-NC ↑									
IMDb	0.463	0.497	0.496	0.442	0.420	0.489	0.400	0.403	0.436
SST2	0.336	0.384	0.384	0.435	0.466	0.374	0.314	0.458	0.431
BoolQ	0.471	0.490	0.491	0.474	0.403	0.498	0.468	0.475	0.497
TellMeWhy	0.632	0.612	0.612	0.564	0.592	0.608	0.632	0.570	0.593
WikiBio	0.633	0.600	0.482	0.578	0.615	0.481	0.765	0.604	0.645
Avg	0.507	0.517	0.493	0.499	0.499	0.490	0.516	0.502	0.520

Table 1: Faithfulness evaluation with AUC of π-Soft-NC/NS curves for (a) Llama and (b) Mistral. Best / 2nd / 3rd are highlighted on the **Avg** row. Abbr.: Attn = Attention, DL = DeepLIFT, I×G = Input×Gradients, IG = Integrated Gradients, Rnd = Random, Sal = Saliency, VZ = Value Zeroing, RG = ReAGent.

4.4 Quantitative Results

Diagnosis of original Soft-NC/NS. Before comparing methods under the proposed metrics, we first examine the original Soft-NC/NS of Zhao and Shan (2024). As shown in Tab. 2, raw Soft-NC/NS compares methods under substantially different expected numbers of retained tokens. For example, Random and Grad-ELLM retain many more tokens on average ($\mathbb{E}[R] = 80$ and 62) than sparse gradient-based methods such as DeepLIFT, IG, and Saliency ($\mathbb{E}[R] \approx 18$–21). In other words, such evaluation is like erroneously comparing results with retaining probability $\pi = 0.1$ to those with $\pi = 0.4$ in Fig. 3. Therefore, higher Soft-NC/NS scores reflect more words involved in the perturbation rather than more faithful attribution, confirming the need to control the retaining probability.

Metric	Attn	DL	I×G	IG	Rnd	Sal	VZ	Ours
Soft-NS↑	0.052	0.087	0.086	0.089	0.132	0.087	0.032	0.138
Soft-NC↑	0.004	0.006	0.006	0.005	0.005	0.006	0.004	0.013
$\mathbb{E}[R]$	17	20	19	18	80	21	8	62

Table 2: Faithfulness evaluation of *original* Soft-NC/NS on IMDb with Llama. No transformations are applied to the attribution scores. $\mathbb{E}[R]$ denotes the expected number of retained words. The abbreviations are the same as in Table 1.

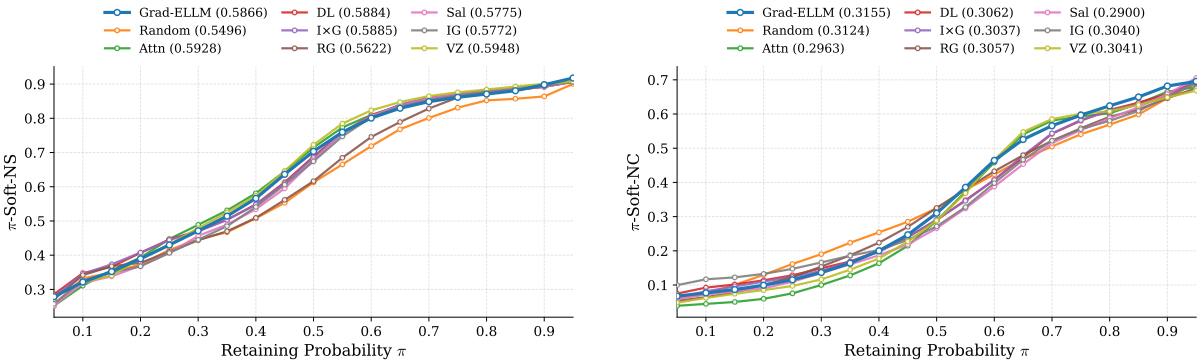
Evaluation under π-Soft-NC/NS. Tab. 1 reports the AUC of π-Soft-NC/NS curves. Grad-ELLM achieves the best average π-Soft-NC on both Llama and Mistral, indicating strong comprehensiveness-oriented faithfulness. On π-Soft-NS, no clear method dominates, with only Input×Gradient in the top-3 for both models. Fig. 3 shows an example set of curves on one dataset. For π-Soft-NS, most methods see advantage over Random in the range $\pi \in [0.5, 0.65]$. For π-Soft-NC, some methods see advantage over Random after when $\pi \geq 0.6$.

Insights Our results in Tab. 1 reveal a sufficiency-comprehensiveness asymmetry: Grad-ELLM is strongest on π-Soft-NC but not always on π-Soft-NS, indicating stronger broad evidence coverage than precise orderings of the top attributions. Meanwhile, sparse attribution methods, like Input×Gradient perform well on π-Soft-NS, indicating that they more precisely identify the most important words. However, it should be noted that the attribution methods only show advantage over Random attributions for larger retaining probability $\pi > 0.5$. It is unclear whether this is due to the need of contextual information to perform the task (i.e., more words need to be retained), or that the top-ranked attributions are misordered.

Finally, we further examine whether the proposed π-Soft metrics merely favor dense attribution maps in Appendix J. The attribution-value distributions show that density alone is insufficient: Random produces the most uniform scores but performs poorly under π-Soft metrics. This suggests that π-Soft-NC/NS reward calibrated selectivity rather than density alone.

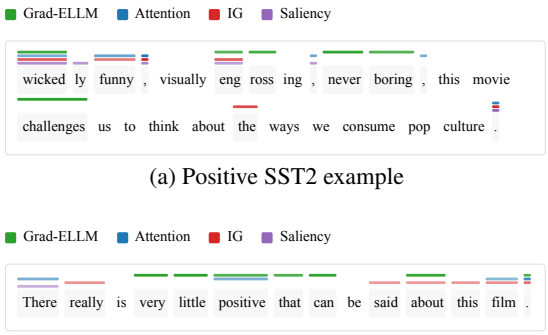
4.4.1 Qualitative Results

Fig. 4 provides qualitative examples of token-level attributions on sentiment classification tasks. For readability, we visualize whether each token is selected among the top 10% attribution scores by each method, rather than showing continuous attribution values. The selected methods are compared on the same input, and colored bars above each



(a) π-Soft-NS vs. Retaining Probability (b) π-Soft-NC vs. Retaining Probability

Figure 3: Proposed π-Soft-NC/NS vs. Retaining Probability on SST2 for different XAI methods with Llama.



(a) Positive SST2 example

(b) Negative IMDb example

Figure 4: Qualitative comparison of token-level attributions on sentiment classification examples. Colored bars above tokens indicate the top 10% attribution scores by each method. For the long IMDb review, we show the key sentence containing the decisive sentiment phrase.

token indicate the tokens selected by different attribution methods.

In the positive example, Grad-ELLM selects sentiment-bearing tokens and phrases such as "engrossing" and "never boring", which jointly support the positive prediction. Attention and gradient-based baselines also select some relevant tokens, but their selected tokens are less consistently aligned with the main positive sentiment cues.

In the negative example, the review is much longer, so we show the key sentence containing the decisive sentiment phrase. Grad-ELLM highlights the phrase "very little positive", which is crucial because the word "positive" alone is misleading without the preceding negation-like modifier. In contrast, other methods tend to focus on isolated tokens such as "positive" or nearby tokens, and may miss the compositional phrase that explains the negative prediction. The full example and additional examples are provided in Appendix F.

5 Conclusion

In this work, we studied faithfulness evaluation for decoder-only LLM attributions. We first identified a bias in the existing Soft-NC/NS metrics (Zhao and Shan, 2024), where different attribution score distributions can lead to different expected amounts of retained input information. To address this issue, we proposed π-Soft-NC and π-Soft-NS, which compare attribution methods under a controlled target retaining probability. We further introduced Grad-ELLM, a gradient-based attribution method that combines gradient-derived channel importance with attention-derived token importance for autoregressive next-token prediction. Experiments across classification and generation tasks show that Grad-ELLM achieves strong comprehensiveness-oriented faithfulness under π-Soft-NC. Meanwhile, there is no clearly dominant method when evaluating with the π-Soft-NS. Compared to Random attributions, most methods only see advantage for larger retaining probability ($\pi > 0.5$), which indicates gaps in both the attribution methods themselves and our understanding of the role of context in the task. We hope that our proposed faithfulness metric and protocol provide a rigorous framework for evaluating XAI attribution methods for LLMs, leading to future progress in the field.

6 Limitations

Grad-ELLM is designed to produce calibrated attribution distributions rather than extremely sparse token rankings. This design is well aligned with soft distributional metrics such as π-Soft-NC/NS, but it may be less aligned with ranking-based hard perturbation metrics such as insertion/deletion (Petsiuk et al., 2018), where the ordering of a few top-ranked tokens dominates the final AUC. Therefore, Grad-ELLM should not be interpreted as universally optimal under all faithfulness metrics; instead, its strength lies in providing a more complete and calibrated attribution distribution.

Furthermore, our study focuses exclusively on faithfulness and does not include evaluations of plausibility, which assesses how intuitive and human-understandable the explanations are. Plausibility is inherently subjective, and quantifying it poses significant challenges—particularly when comparing dense heatmaps (which provide a holistic view of token contributions) to sparse ones (which emphasize focal points). Without standardized, objective metrics or large-scale user studies, it is difficult to determine which type offers better interpretability in practice.

Grad-ELLM also relies on access to internal model states, such as gradients and attention layers, restricting its use to open-source LLMs like Llama and Mistral. Extending it to proprietary models or integrating it with global interpretability techniques in future research could address these gaps.

Another limitation is that π-Soft-NC/NS still rely on embedding-level perturbations. Although controlling the retaining probability improves fairness across attribution methods, the perturbed representations may still differ from naturally occurring text inputs, which is a general concern in perturbation-based interpretability (Ancona et al., 2018; Bastings and Filippova, 2020; Yin et al., 2022). Future work could combine retained-information control with more semantically constrained perturbations.

References

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. 2021. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2882–2887.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

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.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29.

Mohsen Fayyaz, Fan Yin, Jiao Sun, and Nanyun Peng. 2024. Evaluating human alignment and model faithfulness of llm rationale. *Preprint*, arXiv:2407.00219.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, 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.

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. 2024. Faithscore: Fine-grained evaluations of hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5042–5063.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, Online. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Haoyan Luo and Lucia Specia. 2024. From understanding to utilization: A survey on explainability for large language models. *arXiv preprint arXiv:2401.12874*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. 2023. Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400, Dubrovnik, Croatia. Association for Computational Linguistics.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 151. BMVA Press.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, Oskar van der Wal, Malvina Nissim, and Arianna Bisazza. 2023. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Michael Veale and Reuben Binns. 2017. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society*, 4(2):2053951717743530.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Fan Yin, Zhouxing Shi, Cho-Jui Hsieh, and Kai-Wei Chang. 2022. On the sensitivity and stability of model interpretations in nlp. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2631–2647.

Chenyang Zhao, Kun Wang, Janet H Hsiao, and Antoni B Chan. 2025. Grad-eclip: Gradient-based visual and textual explanations for clip. *arXiv preprint arXiv:2502.18816*.

Chenyang Zhao, Kun Wang, Xingyu Zeng, Rui Zhao, and Antoni B Chan. 2024a. Gradient-based visual explanation for transformer-based clip. In *International Conference on Machine Learning*, pages 61072–61091. PMLR.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024b. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Zhixue Zhao and Boxuan Shan. 2024. Reagent: A model-agnostic feature attribution method for generative language models. *arXiv preprint arXiv:2402.00794*.

A Implementation Details

We use Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 from Hugging Face Transformers (Wolf et al., 2020). A larger Llama-3.1-70B-Instruct evaluation is reported in the Appendix G. Baselines are implemented with Inseq (Sarti et al., 2023). We set `top_p=1` for deterministic generation across attribution methods. Experiments with 7B/8B models are run on a single 48GB NVIDIA RTX6000 Ada GPU; the 70B model is evaluated on a single 80GB A100 with 4-bit quantization. Results are averaged over 15 Monte Carlo samples.

We generated heatmaps for all compared attribution methods using the Inseq library. Given an input prompt, we first obtained the model prediction and then computed token-level attribution scores for the input tokens with respect to the generated output. All Inseq-based methods were used with their default hyperparameters, except for Integrated Gradients and ReAGent. For Integrated Gradients, we used 16 integration steps with an internal batch size of 8, which provides a practical trade-off between attribution stability and GPU memory consumption. Since ReAGent relies on iterative probing and is significantly more expensive than gradient- or attention-based methods, we adopted a computationally feasible setting with `keep_top_n=3`, `stopping_condition_top_k=3`, `replacing_ratio=0.25`, `max_probe_steps=32`, and `num_probes=2`. For each sample, the generated attribution scores were aggregated to obtain the final heatmap used in our evaluation.

B Derivation of Grad-ELLM

We provide the derivation of Grad-ELLM used in Sec. 3.3. We focus on explaining the contribution of input tokens to l_t, the logit of the next generated token y_t. Analogous to Grad-ECLIP, we decompose the decoder to relate l_t to intermediate features (see Fig. 2). We first consider the last layer ($k = 0$) where the final hidden state for the last token is $\mathbf{z}_t^{(0)}$. The logit $l_t = \mathcal{LP}(\mathbf{z}_t^{(0)})$, where $\mathcal{LP}$ is the linear projection to vocabulary space. Approximating linearity,

$$l_t = \mathcal{LP}(\mathbf{z}_t^{(0)}) = \mathcal{LP}(\mathbf{o}_t^{(0)} + \mathbf{z}_t^{(1)}) \quad (12)$$

$$\approx \mathcal{LP}(\mathbf{o}_t^{(0)}) + \mathcal{LP}(\mathbf{z}_t^{(1)}), \quad (13)$$

where $\mathbf{o}_t^{(0)}$ is the t-th token in $\mathbf{o}^{(0)}$, and

$$\mathbf{o}_t^{(0)} = \mathcal{A}(\mathbf{z}^{(1)})_t = \sum_i \mathrm{softmax}(\frac{\mathbf{q}_t \cdot \mathbf{k}_i^\top}{\sqrt{d}})\mathbf{v}_i, \quad (14)$$

and $\mathcal{A}$ represents the self-attention layer. Thus we obtain the approximation of logit l_t recursively,

$$l_t \approx \mathcal{LP}(\mathbf{o}_t^{(0)}) + \mathcal{LP}(\mathbf{o}_t^{(1)}) + \cdots + \mathcal{LP}(\mathbf{o}_t^{(N-1)}) + \mathcal{LP}(\mathbf{z}_t^{(N)}) \triangleq \sum_k l_t^{(k)}, \quad (15)$$

as an aggregation of features from each layer.

Following Zhao et al. (2025), and looking at the last transformer layer as an example, we approximate the logit of the next predicted token as a weighted combination of the channel features in $\mathbf{o}_t$. For the last layer ($k = 0$),

$$l_t^{(0)} = \mathcal{LP}(\mathbf{o}_t^{(0)}) = \sum_c \mathcal{LP}(\mathbf{o}_t)[c]^{(0)} \triangleq f(\mathbf{o}_t), \quad (16)$$

where we have defined the logit of the next generated token as a function of $\mathbf{o}_t$, i.e., $f(\mathbf{o}_t)$. We denote the linear approximation of the logit as $\tilde{f}$,

$$f(\mathbf{o}_t) \approx \tilde{f}(\mathbf{o}_t) \triangleq \sum_c w_c o_t[c] = \mathbf{w}\mathbf{o}_t^\top \tag{17}$$

The linear weights $\mathbf{w}$ are obtained by matching the first derivatives of f and its linear approximation:

$$\mathbf{w} = \underset{w}{\operatorname{argmin}} \|f'(\mathbf{o}_t) - \tilde{f}'(\mathbf{o}_t)\|^2 \tag{18}$$

$$= \operatorname{argmin} \|\tfrac{\partial f}{\partial \mathbf{o}_t} - \mathbf{w}\|^2, \tag{19}$$

and thus we obtain the solution $\mathbf{w}^* = \frac{\partial f}{\partial \mathbf{o}_t}$. Finally, combining with (8) we have the approximation:

$$f(\mathbf{o}_t) \approx \sum_c w_c o_t[c] \tag{20}$$

$$= \sum_c \tfrac{\partial f(\mathbf{o}_t)}{\partial \mathbf{o}_t} \sum_i \operatorname{softmax}(\tfrac{\mathbf{q}_t \cdot \mathbf{k}_i^\top}{\sqrt{d}}) \mathbf{v}_{ic} \tag{21}$$

$$= \sum_i \Big[\sum_c \underbrace{\tfrac{\partial f(\mathbf{o}_t)}{\partial \mathbf{o}_t}}_{w_c} \underbrace{\operatorname{softmax}(\tfrac{\mathbf{q}_t \cdot \mathbf{k}_i^\top}{\sqrt{d}})}_{\lambda_i} \mathbf{v}_{ic} \Big], \tag{22}$$

where $\mathbf{v}_{ic}$ is the c-th channel of $\mathbf{v}_i$. Thus the attribution for the i-th token is

$$H_i = \operatorname{ReLU}(\sum_c w_c \lambda_i \mathbf{v}_{ic}), \tag{23}$$

where ReLU is used to focus only on tokens with positive influence on the logit value. In this way we decompose the attribution into two parts: channel weights w_c and token weights λ_i. Following Zhao et al. (2024a), the attribution map will be obtained by $\mathbf{H} = [H_i]_i$ using the last layer value v as the feature map. Based on (15), we can select the desired number of layers to aggregate information from different layers to obtain the heatmap.

As with Zhao et al. (2024a), we also apply 0-1 normalization on the similarities $[\mathbf{q}_t \mathbf{k}_i^\top]_i$ and use it to replace the original softmax, i.e., $\lambda_i \approx \Phi(\mathbf{q}_t \mathbf{k}_i^\top)$, where Φ is the 0-1 normalization function applied over the set of similarities. Without this loosening step, the softmax operation will produce sparser heatmaps, due to the peakiness of softmax. The loosening step helps to reveal unattended input tokens that are similar to the attended token, which likely contain similar information. This design makes Grad-ELLM different from attribution methods that aim to produce extremely sparse token rankings. By combining attention-derived token weights with gradient-derived channel weights, Grad-ELLM produces a continuous attribution distribution over the input.

C Dataset Details

More detailed descriptions of the datasets:

- **IMDb** is introduced by (Maas et al., 2011) which is a widely used benchmark for binary sentiment classification. It consists of 50,000 movie reviews collected from the Internet Movie Database (IMDb), each labeled as either positive or negative. We randomly sampled 1000 positive examples and 1000 negative examples from them.
- **SST2** is also a sentiment classification task introduced by Socher et al. (2013), which consists of independent single sentences, many of which contain complex grammatical structures, negation, and irony, making classification more challenging compared with IMDb. We also randomly sampled 1000 entries from each of the positive and negative samples.
- **BoolQ** is a dataset from the ERASER (DeYoung et al., 2020) benchmark, which tests whether a model can read a paragraph and correctly answer a factual yes/no question about it, emphasizing deep language understanding over superficial cues.
- **TellMeWhy** is a generation dataset for answering questions in narratives. We follow the dataset setting in (Zhao and Shan, 2024; Lal et al., 2021).
- **WikiBio** is a dataset consisting of Wikipedia biographies. Following Zhao and Shan (2024); Manakul et al. (2023), we take the first two sentences as a prompt.

D Prompt Details

The full prompts are shown in Tab. 3. For Llama, we add some special tokens at the end of the input text to ensure that the model outputs text in the desired format. These special tokens are: "<|start_header_id|> assistant <|end_header_id|> \n". These tokens correspond to the assistant header expected by the Llama chat template. If we don't force the model to output them, it might not output them according to our format, which could lead to incorrect output during deletion and insertion evaluation. However, this requirement does not apply to Mistral.

Dataset	Prompt Construction Code

IMDb & SST2	`messages = [`
	`  {`
	`  "role": "system",`
	`  "content": "You are a helpful sentiment classifier. Please help to do the sentiment classification`
	`  of the given text and respond ONLY with the single word Positive or Negative.",`
	`  },`
	`  {"role": "user", "content": f"Text: {text}"},`
	`  ]`

BoolQ	`messages = [`
	`  {`
	`  "role": "system",`
	`  "content": "You are a factual question answering system. "`
	`  "Determine if the given passage supports the question being true. "`
	`  "Respond ONLY with the single word 'true' or 'false' in lowercase, "`
	`  "without any punctuation or additional text."`
	`  },`
	`  {`
	`  "role": "user",`
	`  "content": "Passage: {passage}\n\nQuestion: {question}".format(`
	`  passage=passage_text,`
	`  question=question_text`
	`  )`
	`  }`
	`  ]`

TellMeWhy & WikiBio	`messages = [`
	`  {`
	`  "role": "user",`
	`  "content": text`
	`  }`
	`  ]`

Table 3: Detailed prompts for all datasets

E Evaluation with Insertion/Deletion

For completeness, we also compute the classical Deletion and Insertion (Petsiuk et al., 2018) metrics for classification tasks. Inspired by DeYoung et al. (2020), to more robustly evaluate the performance of attribution methods, we evaluate the change in flip probability rather than the change in prediction probability. In our experiment setting, we set each perturbation step to 5% of the tokens and record results for 20 steps.

Higher AUC for Deletion is better, since ideally removing words will yield large flip probability (large performance degradation). Conversely, lower AUC for Insertion is better, since ideally adding words will yield a reduction in flip probability (large performance gain). It is worth noting that if the change in prediction probability is the evaluation target, Value Zeroing should theoretically be the upper bound for Deletion and insertion experiments, as it directly measures the causal effect of zeroing features on the model's prediction. However, within the evaluation framework of flip probability, this upper bound property may be weakened, because flip probability focuses on crossing class boundaries rather than the absolute change in probability magnitude.

E.1 Results

The Insertion/Deletion AUC results are presented in Tabs.4 and 5. In the deletion experiments, our method achieved the second-best result with Llama and the third-best result with Mistral. However, our method yielded slightly worse results in the insertion experiments. Through in-depth analysis, we found that our heatmap is denser compared to other methods, which puts us at a disadvantage in order-based evaluations such as deletion and insertion, because we are more susceptible to small noise that swaps the order of words. On the other hand, as presented in the main paper, our π-Soft-NC/NS metrics reveal that our method has better overall importance distributions.

F Additional Qualitative Results

We provide additional qualitative examples to complement Fig. 4. Since IMDb reviews can be substantially longer than SST2 sentences, the main paper only shows the key sentence from the negative IMDb example. Here, we show the full IMDb review corresponding to Fig. 4(b), as well as another negative IMDb example with a similar attribution pattern.

Fig. 7 shows the full review for the negative IMDb example in the main paper. The key phrase "very little positive" appears in the full context, and Grad-ELLM still selects the phrase-level evidence

Deletion↑	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	ReAGent	Ours (w/ | w/o)
IMDb	0.2572	0.1956	0.1929	0.1690	0.1772	0.1681	0.2618	0.1820	0.2153 | 0.2520
SST2	0.3310	0.3050	0.3058	0.2821	0.2868	0.2741	0.3462	0.2987	0.3251 | 0.3383
BoolQ	0.2393	0.2958	0.2956	0.2852	0.2292	0.2955	0.2490	0.2299	0.2673 | 0.2621
Average	0.2758	0.2655	0.2648	0.2454	0.2311	0.2459	0.2857	0.2369	0.2692 | 0.2841
Insertion↓	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	ReAGent	Ours (w/ | w/o)
IMDb	0.0847	0.1242	0.1294	0.1275	0.1718	0.1358	0.0832	0.1706	0.1336 | 0.1021
SST2	0.1816	0.2022	0.2022	0.2174	0.2579	0.2217	0.1754	0.2532	0.2160 | 0.1957
BoolQ	0.1885	0.0991	0.1005	0.1137	0.2087	0.0995	0.1902	0.2106	0.1740 | 0.1764
Average	0.1516	0.1418	0.1440	0.1529	0.2128	0.1523	0.1496	0.2115	0.1745 | 0.1581

Table 4: Faithfulness evaluation of Deletion and Insertion on IMDb, SST2, and BoolQ with Llama. The best performance, second place, and third place are marked in red, blue, and green respectively. w/ denotes loosened attention, and w/o denotes original attention.

Deletion↑	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	ReAGent	Ours (w/ | w/o)
IMDb	0.2719	0.1967	0.1951	0.1726	0.1892	0.1898	0.2415	0.1795	0.2296 | 0.2266
SST2	0.3550	0.2821	0.2839	0.3013	0.2903	0.2828	0.3486	0.2857	0.3167 | 0.3226
BoolQ	0.2776	0.3042	0.3023	0.2812	0.2339	0.3055	0.2579	0.2339	0.2658 | 0.2606
Average	0.3015	0.2610	0.2604	0.2517	0.2378	0.2593	0.2826	0.2330	0.2707 | 0.2699
Insertion↓	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	ReAGent	Ours (w/ | w/o)
IMDb	0.0861	0.1079	0.1074	0.1189	0.1849	0.1062	0.0973	0.1806	0.1200 | 0.1273
SST2	0.1863	0.2127	0.2121	0.2073	0.2652	0.2089	0.1899	0.2532	0.2023 | 0.2047
BoolQ	0.1556	0.1067	0.1079	0.1310	0.2226	0.1044	0.1974	0.2191	0.1983 | 0.1953
Average	0.1427	0.1424	0.1425	0.1524	0.2242	0.1398	0.1615	0.2177	0.1735 | 0.1757

Table 5: Faithfulness evaluation of Deletion and Insertion on IMDb, SST2, and BoolQ with Mistral. The best performance, second place, and third place are marked in red, blue, and green respectively. w/ denotes loosened attention, and w/o denotes original attention.

rather than only the isolated token "positive". This suggests that Grad-ELLM can capture compositional sentiment cues within a longer review.

Fig. 8 shows another negative IMDb example. In this review, Grad-ELLM highlights multiple strongly negative phrases, such as "nothing positive", "terrible piece of film", and related negative expressions. This example further illustrates that the method does not merely select sentiment words in isolation, but can highlight broader phrases that support the model's negative prediction.

G Results on Larger Models

We also conduct experiments with Llama-3.1-70B-Instruct to evaluate whether the proposed metrics and Grad-ELLM remain effective on a larger model. The results are shown in Tab. 7. Overall, the 70B results show a pattern consistent with the main 7B/8B experiments: Grad-ELLM is especially strong under the comprehensiveness-oriented π-Soft-NC metric, achieving the best average π-Soft-NC AUC across datasets. This suggests that Grad-ELLM's attribution distribution continues to provide broad coverage of important evidence when scaling to a larger LLM.

On π-Soft-NS, Grad-ELLM remains competitive but is not the top-performing method on average. This again indicates that sufficiency-oriented evaluation and comprehensiveness-oriented evaluation capture different attribution behaviors. Sparse methods such as Attention, Integrated Gradients, or Saliency can sometimes better preserve the prediction with a smaller subset of retained tokens, while Grad-ELLM is more effective at identifying evidence whose removal causes a larger distributional shift. These results further support our main argument that faithful attribution should be evaluated along multiple axes rather than by a single metric.

H Computational Complexity

We analyze the average attribution time per single sample for different methods on the IMDb dataset. The results are shown in Tab. 6. As shown in the table, our gradient-based method is faster than most of the compared methods while achieving competitive results. Note that ReAGent and Value Zeroing are much slower because they require multiple token replacements and forward passes. More specifically, Value Zeroing requires over 45.2 seconds per sample, while our method only takes 0.32 seconds.

Method	Llama-8B	Mistral-7B
Attention	**0.31**	**0.32**
DeepLIFT	0.95	1.35
Input×Gradient	0.63	0.80
Integrated Gradients	6.45	9.05
Saliency	0.63	0.81
Value Zeroing	45.20	47.80
ReAGent	17.50	18.70
Ours	0.32	0.38

Table 6: Average Time per Sample (Seconds) for generating heatmaps on IMDb (Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3). Times averaged over evaluation set.

π-Soft-NS↑	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	Ours
IMDb	0.7348	0.6937	0.6982	0.7447	0.7069	0.7036	0.7205	0.7152
SST2	0.6763	0.6230	0.6296	0.6839	0.5949	0.6299	0.6432	0.6586
BoolQ	0.5090	0.5235	0.5232	0.5210	0.4273	0.5374	0.5119	0.4834
TellMeWhy	0.6369	0.6277	0.6320	0.5723	0.4513	0.6415	0.6004	0.6152
WikiBio	0.5369	0.4808	0.4899	0.4812	0.3687	0.5009	0.4872	0.4934
Average	0.6187	0.5897	0.5945	0.6006	0.5098	0.6026	0.5926	0.5931
π-Soft-NC↑	Attention	DeepLIFT	Input×Gradients	Integrated Gradients	Random	Saliency	Value Zeroing	Ours
IMDb	0.2112	0.1607	0.1699	0.1849	0.1822	0.1654	0.1880	0.2068
SST2	0.3337	0.2619	0.2662	0.3124	0.3082	0.2803	0.2746	0.3228
BoolQ	0.4718	0.4622	0.4640	0.4819	0.4962	0.4892	0.4820	0.5343
TellMeWhy	0.4769	0.3985	0.4004	0.3510	0.3829	0.4203	0.3752	0.4305
WikiBio	0.5853	0.5685	0.5617	0.4976	0.5135	0.5708	0.5694	0.6060
Average	0.4157	0.3703	0.3724	0.3655	0.3766	0.3852	0.3778	0.4200

Table 7: Faithfulness evaluation with AUC of π-Soft-NC/NS curves for Llama-3.1-70B-Instruct. The best performance, second place, and third place are marked in red, blue, and green respectively.

I Layer Aggregation Analysis

Grad-ELLM aggregates attribution information from multiple decoder layers, as described in Eq. (5). The number of aggregated layers controls how much intermediate attribution evidence is incorporated into the final heatmap. Since attribution behavior can vary across datasets and model families, we do not fix this number to all decoder layers. Instead, we perform a lightweight search over a small candidate set of layer numbers and select the configuration that performs best for each dataset and model.

Specifically, we evaluate Grad-ELLM with the number of aggregated layers selected from $\{1, 4, 8, 16, 32\}$. For each setting, we compute the AUC of the π-Soft-NS and π-Soft-NC curves using the same evaluation protocol as in the main experiments. The purpose of this analysis is to examine how sensitive Grad-ELLM is to the layer aggregation depth and to justify the layer configurations used in the main results.

As shown in Fig. 5, the optimal number of aggregated layers is not always the maximum number of layers. This indicates that attribution evidence is distributed differently across layers depending on the dataset and metric. Using too few layers may miss useful intermediate evidence, while aggregating too many layers can also introduce noisy or less task-relevant attribution signals. Therefore, in the main experiments, we select the layer aggregation depth from a small candidate set rather than always using all layers.

J Attribution Distribution Analysis

To examine whether π-Soft-NC/NS simply favor dense attribution maps, we visualize the distribution of attribution values across different methods. Fig. 6 shows that Grad-ELLM produces a more evenly distributed attribution map than sparse gradient-based methods such as DeepLIFT and Saliency. However, density alone does not explain faithfulness performance: Random produces the most uniform attribution distribution but performs poorly under π-Soft metrics.

This suggests that the proposed metrics do not merely reward dense heatmaps. Instead, strong π-Soft performance requires calibrated selectivity, where attribution scores distribute probability mass broadly enough to cover relevant evidence while still distinguishing informative tokens from uninformative ones.

K Fixed-Routing Causal Evaluation

Here we present the results using the *fixed-routing* protocol. In particular, for each input, we first run the model on the original unperturbed sequence and cache the attention maps. During the evaluation of softly perturbed inputs, we reuse the cached attention maps, so that the attention routing remains the same as in the original forward pass. Under this protocol, changes in the output distribution are mainly caused by changes to the value/content information carried by the perturbed tokens, rather than the potential influence of attention reallocation.

The results are presented in Table 8 and Fig. 9. Input×Gradient, Saliency and DeepLift perform well under the π-Soft-NS metric. This is likely due to their use of input-level gradients that measure the effect of an input token on the output. On the other hand, there is no clearly dominant method for π-Soft-NC under fixed-routing. Note that several methods consistently outperform raw attention under the fixed-routing protocol. This indicates that attention itself is lacking as an attribution method – some attended words do not have large affects on the output.

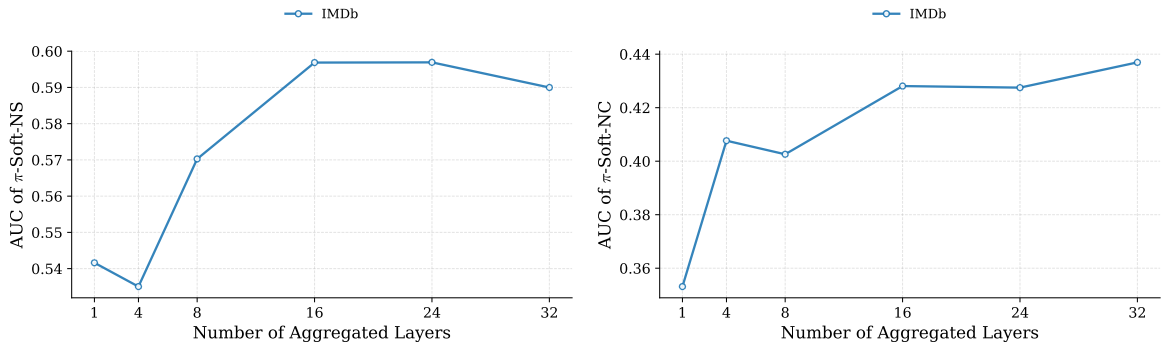


Figure 5: Layer aggregation analysis for Grad-ELLM on IMDb with Llama. The x-axis denotes the number of aggregated decoder layers, and the y-axis denotes the AUC of the corresponding π-Soft metric. Left: AUC of π-Soft-NS. Right: AUC of π-Soft-NC.

(a) Llama: AUC of π-Soft-NC/NS under fixed-routing evaluation (↑)

Dataset	Attn	DL	I×G	IG	Rnd	Sal	VZ	RG	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.649	0.633	0.632	0.629	0.582	0.624	0.645	0.596	0.644
SST2	0.512	0.509	0.510	0.503	0.433	0.490	0.524	0.464	0.517
BoolQ	0.386	0.455	0.452	0.425	0.366	0.450	0.389	0.364	0.412
TellMeWhy	0.644	0.668	0.665	0.647	0.520	0.663	0.649	0.555	0.625
WikiBio	0.653	0.654	0.652	0.649	0.506	0.648	0.663	0.550	0.624
Avg	0.569	0.584	0.582	0.571	0.482	0.575	0.574	0.506	0.566
AUC π-Soft-NC ↑									
IMDb	0.348	0.314	0.312	0.318	0.309	0.309	0.339	0.299	0.320
SST2	0.505	0.521	0.519	0.524	0.504	0.501	0.533	0.476	0.519
BoolQ	0.571	0.704	0.698	0.644	0.691	0.692	0.576	0.637	0.675
TellMeWhy	0.562	0.572	0.566	0.527	0.397	0.566	0.572	0.395	0.484
WikiBio	0.593	0.521	0.519	0.543	0.428	0.511	0.638	0.411	0.502
Avg	0.516	0.526	0.523	0.511	0.466	0.516	0.532	0.443	0.500

(b) Mistral: AUC of π-Soft-NC/NS under fixed-routing evaluation (↑)

Dataset	Attn	DL	I×G	IG	Rnd	Sal	VZ	RG	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.612	0.627	0.626	0.626	0.584	0.631	0.579	0.588	0.622
SST2	0.502	0.499	0.499	0.517	0.485	0.498	0.503	0.478	0.517
BoolQ	0.455	0.482	0.481	0.469	0.433	0.484	0.445	0.426	0.446
TellMeWhy	0.511	0.520	0.521	0.485	0.400	0.522	0.512	0.423	0.469
WikiBio	0.629	0.628	0.629	0.609	0.480	0.632	0.627	0.519	0.583
Avg	0.542	0.551	0.551	0.541	0.476	0.553	0.533	0.487	0.527
AUC π-Soft-NC ↑									
IMDb	0.404	0.411	0.410	0.404	0.356	0.406	0.388	0.321	0.405
SST2	0.711	0.728	0.730	0.779	1.031	0.891	0.719	1.262	1.076
BoolQ	0.465	0.509	0.505	0.475	0.460	0.517	0.435	0.409	0.465
TellMeWhy	0.800	0.789	0.798	0.766	0.678	0.789	0.817	0.623	0.825
WikiBio	0.667	0.611	0.611	0.582	0.485	0.616	0.686	0.494	0.547
Avg	0.609	0.609	0.611	0.601	0.602	0.644	0.609	0.622	0.664

Table 8: Fixed-routing protocol with AUC of π-Soft-NC/NS curves for (a) Llama and (b) Mistral. Attention maps are cached from the original input and reused when evaluating softly perturbed inputs. This protocol tests whether the token content selected by each attribution method remains causally important under the original attention routing. Best / 2nd / 3rd are highlighted on the **Avg** row. Abbr.: Attn = Attention, DL = DeepLIFT, I×G = Input×Gradients, IG = Integrated Gradients, Rnd = Random, Sal = Saliency, VZ = Value Zeroing, RG = ReAGent.

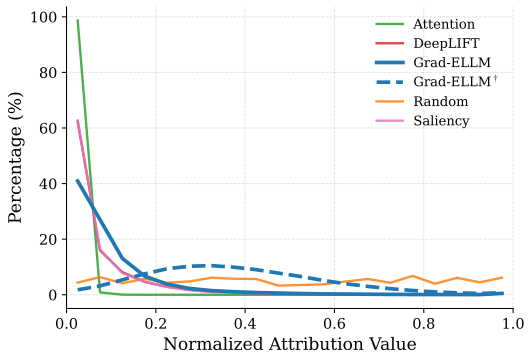


Figure 6: Attribution-value distributions across different methods with Llama on IMDb. Grad-ELLM† denotes the variant with loosened attention normalization. Random is dense and nearly uniform but performs poorly under π-Soft metrics, indicating that density alone does not guarantee high faithfulness.

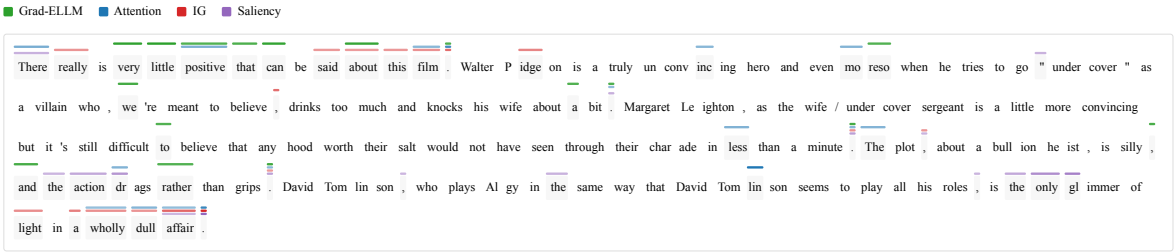


Figure 7: Full review for the negative IMDb example shown in Fig. 4(b). Colored bars indicate tokens selected among the top 10% attribution scores by each method. The full context shows that the key phrase "very little positive" is part of a longer negative review.

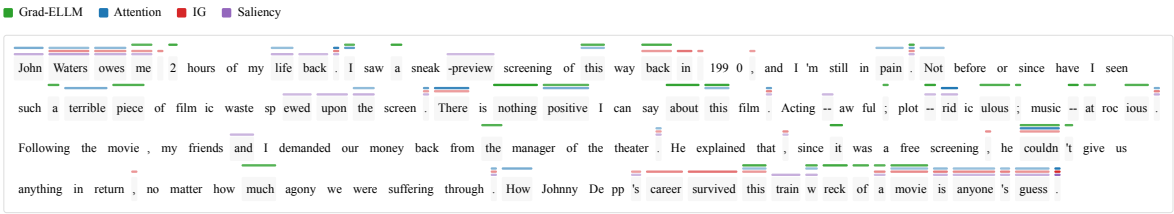
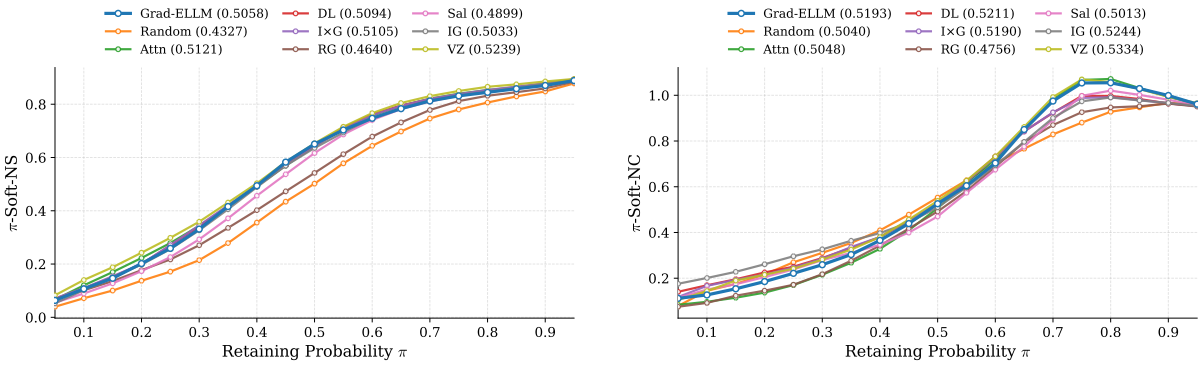


Figure 8: Additional negative IMDb example. Colored bars indicate tokens selected among the top 20% attribution scores by each method.



(a) π-Soft-NS vs. Retaining Probability (b) π-Soft-NC vs. Retaining Probability

Figure 9: Fixed-Routing version of proposed π-Soft-NC/NS vs. Retaining Probability on SST2 for different XAI methods with Llama.