Grad-ELLM: Gradient-based Explanations for Decoder-only LLMs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse tasks, yet their black-box nature raises concerns about transparency and faithfulness. Input attribution methods aim to highlight each input token's contributions to the model's output, but existing approaches are typically model-agnostic, and do not focus on transformer-specific architectures, leading to limited faithfulness. To address this, we propose Grad-ELLM, a gradient-based attribution method for decoder-only transformer-based LLMs. By aggregating channel importance from gradients of the output logit with respect to attention layers and spatial importance from attention maps, Grad-ELLM generates heatmaps at each generation step without requiring architectural modifications. Additionally, we introduce two faithfulneses metrics π-Soft-NC and π-Soft-NS, which are modifications of Soft-NC/NS that provide fairer comparisons by controlling the amount of information kept when perturbing the text. We evaluate Grad-ELLM on sentiment classification, question answering, and open-generation tasks using different models. Experiment results show that Grad-ELLM consistently achieves superior faithfulness than other attribution methods.

1 Introduction

Large language models such as Llama have demonstrated their remarkable ability on multiple downstream tasks, e.g., text generation and sentiment classification (Touvron et al., 2023). 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 accurate and fair decision-making is crucial. Lack of interpretability also poses risks of generating harmful or misleading contents, including biases, hallucinations, and toxic outputs (Binns, 2018).

Explainable AI (XAI) has emerged as a critical area to address these challenges. The task of input attribution aims to identify the parts of the input that contribute most to the model's output, thereby revealing the model's reasoning (Sundararajan et al., 2017; Lei et al., 2016). Many general attribution methods have been proposed (Ribeiro et al., 2016; Sundararajan et al., 2017; Shrikumar et al., 2017; Simonyan et al., 2014), which are designed to work with any model and data type, while others are specifically tailored to language model tasks (Zhao and Shan, 2024; Cohen-Wang et al., 2024; Zhang et al., 2022; Boyle et al., 2025). Meanwhile, there is less related work on using transformer-specific explanations for decoder-only LLMs, e.g., autoregressive LLMs.

To fill this research gap, we propose Grad-ELLM, a gradient-based explanations for LLMs. Inspired by the outstanding performance of Grad-ECLIP (Zhao et al., 2024a) in explaining the transformer-based CLIP (Radford et al., 2021) vision-language model (VLMs), in this paper we extend Grad-ECLIP to explain decoder-only LLMs. Grad-ECLIP generates heat maps by aggregating the intermediate token representation of CLIP with channel and spatial importance weights. However, CLIP is a dual-encoder model trained with contrastive learning. To apply Grad-ECLIP to explain decoder-only LLMs, we use the gradients of the logit of the last token w.r.t the attention layer as the importance for feature channels and use the same loosened attention map for computing token (spatial) importance. Then a heat map is calculated for each LLM generation step.

Attribution methods are evaluated using faithfulness metrics, which measure how well the attribution scores reflect the actual importance of those inputs – the inputs are perturbed according to the at-

tribution scores (e.g., inputs with higher-attribution scores are removed first) and then the change of the model's output is observed. Recently, Zhao and Shan (2024) proposed Normalized Soft Completeness and Soft Sufficiency (Soft-NC/NS) to better evaluate the faithfulness of the entire importance distribution, which is based on soft-perturbations that keep (or remove) words randomly based on the importance distribution. However, we have identified an issue with Soft-NC/NS when comparing two attribution methods – if the attribution scores of the 2 methods have different means, then they will have different expected numbers of kept words (i.e., kept information) in the soft-perturbation, leading to unfair comparisons. To address this, we propose a modification, denoted as π-Soft-NC/NS, which ensures the soft-perturbations have the same amount of kept information.

In summary, our contributions are two-fold: 1) we propose Grad-ELLM and show that it is consistently more faithful than other attribution methods by conducting experiments on 2 types of tasks and 5 datasets from 2 LLM families; 2) we propose new faithfulness metrics that accounts for the amount of information loss/kept during evaluation in order to achieve fairer comparisons.

2 Related Work

Existing research on XAI for LLMs can be broadly categorized into local and global explanation methods, as outlined in recent surveys (Zhao et al., 2024b; Luo and Specia, 2024). Local explanations focus on interpreting individual predictions, such as through input attributions, while global explanations aim to uncover broader model behaviors, including internal representations and mechanisms. Our work primarily builds on local input attribution techniques tailored for decoder-only LLMs, addressing their unique autoregressive nature.

2.1 General Input Attribution Methods

Input attribution methods seek to quantify the contribution of input features (e.g., tokens) to model outputs, and early methods were model-agnostic and originated from computer vision or general ML contexts, but have been adapted to NLP. Gradient-based approaches compute attributions by propagating gradients from the output back to the inputs. Integrated Gradients (IG) (Sundararajan et al., 2017) integrates gradients along a path from a baseline input to the actual input, satisfying axioms like completeness and sensitivity. DeepLIFT (Shrikumar et al., 2017) decomposes the output difference from a reference point by assigning contributions based on neuron activations, offering efficiency over vanilla gradients. Input$\times$Gradient (Simonyan et al., 2014) simply multiplies input values by their gradients, providing a fast approximation but sometimes suffering from saturation issues in deep networks. These methods are computationally lightweight and faithful in encoder-based models like BERT (Devlin et al., 2019), but their application to decoder-only LLMs (e.g., Llama (Touvron et al., 2023), Mistral (Jiang et al., 2023)) is challenging due to the need for per-token attributions during autoregressive generation, leading to accumulated noise and reduced faithfulness (Zhao and Shan, 2024; Anonymous, 2024). Perturbation-based methods alter inputs to observe output changes. LIME (Ribeiro et al., 2016) approximates the model locally with a linear surrogate, sampling perturbations around the input to estimate feature importance. While interpretable, it assumes feature independence and can be sensitive to sampling strategies, potentially destroying semantic coherence in text (Ribeiro et al., 2016).

Our proposed Grad-ELLM builds on gradient-based methods by extending them 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 works address LLM-specific challenges, such as autoregressive generation and long contexts. For instance, Anonymous (2024) evaluate attribution faithfulness on instruction-tuned LLMs, arguing that low downstream accuracy (e.g., in classification) can confound evaluations. Some LLM-focused methods include causal tracing for knowledge localization (Meng et al., 2022), but they are more suited for global interpretability rather than per-input attribution. Other LLM-tailored methods focus on perturbing to explain LLMs. Value Zeroing (Mohebbi et al., 2023) masks or zeros out tokens at each generation step and measures the impact on output representations, providing direct causal insights. However, it requires multiple forward passes per token, making it prohibitively expensive for long sequences in LLMs (e.g., $O(n^2)$ complexity for n tokens) (Mohebbi et al., 2023). Reagent (Zhao and Shan, 2024) uses an auxiliary

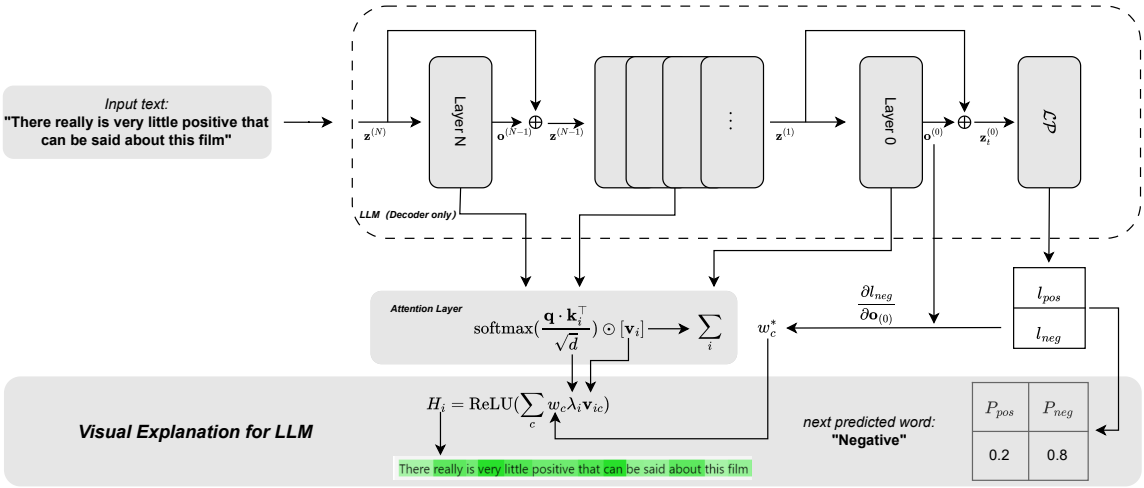


Figure 1: Framework of the proposed Grad-ELLM. An attribution map is generated by aggregating the values $\mathbf{v}$ as feature map in the attention layer with token weight λ_i and channel weight w_c. This example shows the attribution map when generating the word "Negative".

RoBERTa model (Liu et al., 2020) to generate semantically-preserving perturbations, improving robustness over LIME, but it introduces additional computational overhead and may not generalize well across LLM architectures due to the reliance on an external perturber. Other attention-based methods leverage the transformer's self-attention mechanism (Vaswani et al., 2017) as a proxy for explanations. Raw attention weights visualize token interactions (Bahdanau et al., 2015), but studies show they are often weakly correlated with true model reasoning, leading to plausible but unfaithful explanations (Jain and Wallace, 2019; Serrano and Smith, 2019). Extensions like Grad-SAM (Barkan et al., 2021) incorporate gradients into attention maps for better highlighting, yet they still suffer from sparsity and noise in decoder-only models, where attention evolves across generation steps (Abnar and Zuidema, 2020).

For VLMs, Grad-ECLIP (Zhao et al., 2024a) extends Class Activation Mapping (Zhou et al., 2016) to explain CLIP (Radford et al., 2021) by aggregating spatial and channel importance weights from gradients and softened attention maps. It visualizes image-text alignments but is designed for encoder-decoder architectures, limiting direct applicability to autoregressive LLMs without modifications to handle sequential generation.

Unlike these methods, which often require auxiliary models, multiple perturbations, or architectural changes, Grad-ELLM adapts Grad-ECLIP's aggregation of gradients and attention maps to decoder-only LLMs, enabling faithful, per-step attributions without modifications or high computational costs.

2.3 Evaluation of Attribution Faithfulness

Faithfulness is commonly assessed via insertion/deletion curves (Petsiuk et al., 2018), comprehensiveness/sufficiency (DeYoung et al., 2020), and axiom adherence (e.g., completeness) (Sundararajan et al., 2017). Zhao and Shan (2024) proposes Soft-NC and Soft-NS to handle continuous attributions. However, these metrics may unfairly penalize methods with different information loss profiles during perturbation (see §3.3).

Other faithfulness metrics for LLM attributions include: Area Over the Perturbation Curve (AOPC) (Samek et al., 2016), which averages the change in model output probability across multiple perturbations to assess attribution quality; RemOve And Retrain (ROAR) (Hooker et al., 2019), which evaluates by retraining the model after masking highly attributed features and measuring the performance drop; FaithScore (Jing et al., 2024), which quantifies factual consistency in model-generated rationales; Ragas (Es et al., 2024), which measures how grounded outputs are in retrieved contexts (RAG).

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.

3 Methodology

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

3.1 Preliminary on LLMs

Decoder-only LLMs, such as Llama (Touvron et al., 2023) 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 $\{x_1, \ldots, x_m, y_1, \ldots, y_{t-1}\}$.

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:

$$\lambda_i = \text{softmax}(\frac{\mathbf{q}\cdot\mathbf{k}_i^\top}{\sqrt{d}}), \tag{1}$$

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. 1). 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 $\mathbf{z}^{(k-1)} = \mathbf{o}^{(k-1)} + \mathbf{z}^{(k)}$.

3.2 Proposed Grad-ELLM

We focus on explaining the contribution of input tokens to l_t, the logit of the next generated token y_t. Analogous to Grad-CLIP, we decompose the decoder to relate l_t to intermediate features (see Fig. 1). 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,

$$l_t = \mathcal{LP}(\mathbf{z}_t^{(0)}) = \mathcal{LP}(\mathbf{o}_t^{(0)} + \mathbf{z}_t^{(1)}) \tag{2}$$

$$\approx \mathcal{LP}(\mathbf{o}_t^{(0)}) + \mathcal{LP}(\mathbf{z}_t^{(1)}), \tag{3}$$

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

$$\mathbf{o}_t^{(0)} = \mathcal{A}(\mathbf{z}^{(1)})_t = \sum_i \text{softmax}(\frac{\mathbf{q}_t\cdot\mathbf{k}_i^\top}{\sqrt{d}})\mathbf{v}_i, \tag{4}$$

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

$$l_t \approx \mathcal{LP}(\mathbf{o}_t^{(0)}) + \mathcal{LP}(\mathbf{o}_t^{(1)}) + \cdots + \mathcal{LP}(\mathbf{o}_t^{(N-1)}) + \mathcal{LP}(\mathbf{z}_t^{(N)}) \triangleq \sum_k l_t^{(k)}, \tag{5}$$

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 ($k = 0$),

$$l_t^{(0)} = \mathcal{LP}(\mathbf{o}_t^{(0)}) = \sum_c \mathcal{LP}(\mathbf{o}_t)[c]^{(0)} \triangleq f(\mathbf{o}_t), \tag{6}$$

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 $\tilde{f}$,

$$f(\mathbf{o}_t) \approx \tilde{f}(\mathbf{o}_t) \triangleq \sum_c w_c o_t[c] = \mathbf{w}\mathbf{o}_t^\top \tag{7}$$

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

$$\mathbf{w} = \underset{w}{\text{argmin}} \|f'(\mathbf{o}_t) - \tilde{f}'(\mathbf{o}_t)\|^2 \tag{8}$$

$$= \text{argmin}\, \|\frac{\partial f}{\partial \mathbf{o}_t} - \mathbf{w}\|^2, \tag{9}$$

and thus we obtain the solution $\mathbf{w}^* = \frac{\partial f}{\partial \mathbf{o}_t}$. Finally, combining with (1) we have the approximation:

$$f(\mathbf{o}_t) \approx \sum_c w_c o_t[c] \tag{10}$$

$$= \sum_c \frac{\partial f(\mathbf{o}_t)}{\partial \mathbf{o}_t} \sum_i \text{softmax}(\frac{\mathbf{q}_t\cdot\mathbf{k}_i^\top}{\sqrt{d}})\mathbf{v}_{ic} \tag{11}$$

$$= \sum_i \Big[\sum_c \underbrace{\frac{\partial f(\mathbf{o}_t)}{\partial \mathbf{o}_t}}_{w_c} \underbrace{\text{softmax}(\frac{\mathbf{q}_t\cdot\mathbf{k}_i^\top}{\sqrt{d}})}_{\lambda_i} \mathbf{v}_{ic}\Big], \tag{12}$$

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

$$H_i = \text{ReLU}(\sum_c w_c \lambda_i \mathbf{v}_{ic}), \tag{13}$$

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 (5), 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 heat maps, 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.

3.3 π-Soft-NC and π-Soft-NS

In this section, we propose a modification of Soft-NC/NS, denoted as π-Soft-NC/NS, to obtain a more fair and 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,

$$\mathbf{x}_i' = e_i\mathbf{x}_i, \quad e_i \sim \text{Bernoulli}(s_i). \tag{14}$$

Here we assume $s_i \in [0, 1]$, this 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:

$$\Delta\mathbf{P}_{X',t} = H(\mathbf{P}_{X,t}, \mathbf{P}_{X',t}) \tag{15}$$

$$= \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{v}(\sqrt{p_{i,t}} - \sqrt{p_{i,t}'})}, \tag{16}$$

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

$$\text{Soft-NS}(X, x_t, \mathbf{s}) = \frac{\max(0, \Delta\mathbf{P}_{0,t} - \Delta\mathbf{P}_{X',t})}{\Delta\mathbf{P}_{0,t}}, \tag{17}$$

where $\Delta\mathbf{P}_{X',t}$ is effect of the soft-perturbation X', and $\Delta\mathbf{P}_{0,t}$ is effect of the zero baseline (all zero inputs). Likewise, Soft-NC is defined as the relative effect of removing important tokens,

$$\text{Soft-NC}(X, x_t, \mathbf{s}) = \frac{\Delta\mathbf{P}_{\bar{X}',t}}{\Delta\mathbf{P}_{0,t}}, \tag{18}$$

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.

π-Soft-NC/NS. The original version of Soft-NS/NC has a problem that makes it potentially unfair to compare the performance of different attribution methods. Different attribution methods may yield heat maps 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 attribute scores $\mathbf{s}$, the probability of retaining any word is:

$$\hat{\pi} = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[e_i] = \frac{1}{m}\sum_{i=1}^{m}s_i, \tag{19}$$

and the expected number of kept words is $m\hat{\pi}$. It will be unfair to compare the Soft-NS/NC 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.

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 π-Soft-NC/NS as:

$$\begin{aligned} &\pi\text{-Soft-NC}(X, x_t, \mathbf{s}, \pi) = \text{Soft-NC}(X, x_t, \mathbf{s}^\alpha), \\ &\pi\text{-Soft-NS}(X, x_t, \mathbf{s}, \pi) = \text{Soft-NS}(X, x_t, \mathbf{s}^\alpha), \\ &\quad \text{where } \frac{1}{m}\sum_{i=1}^{m} s_i^\alpha = \pi. \end{aligned} \tag{20}$$

In practice, α is solved numerically, e.g., using bisection search.

Using the same target π, then 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.

4 Experiments

In this section we conduct experiments on Grad-ELLM to evaluate its explanation qualitatively and quantitatively and compare with the current SOTA method. We compared 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) layer-specific gradient-based method Layer Gradient×Activation (similar to Layer-GradCAM (Jiang et al., 2021)); 4) perturbation-based Value Zeroing (Mohebbi et al., 2023); 5) a random baseline (Random), which assigns uniform random scores to tokens as a control.

4.1 Datasets

We experiment with three classification datasets and two text generation datasets, following related work (Zhao and Shan, 2024). **IMDb** is a widely used benchmark for binary sentiment classification of movie reviews (Maas et al., 2011). We randomly sampled 1000 positive examples and 1000 negative examples. **SST2** is a sentiment classification task, which consist of challenging single sentences (Socher et al., 2013). We also randomly sampled 1000 positive and 1000 negative samples. **BoolQ** is a dataset from Eraser(DeYoung et al., 2020) benchmark, which tests whether a model can read a paragraph and correctly answer a factual yes/no question. **TellMeWhy** is a generation dataset for answering questions in narratives (Zhao and Shan, 2024; Lal et al., 2021). **WikiBio** is a dataset consisted of Wikipedia biographies, and following (Zhao and Shan, 2024; Manakul et al., 2023), we take the first two sentences as a prompt.

4.2 Faithfulness Metrics

We mainly use our proposed π-Soft-NC and π-Soft-NS 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 π and each text input, the two metrics are calculated using 3 samples of soft-perturbed text. 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 task, we evaluate the **sequence-level faithfulness** – we evaluate π-Soft-NS/NC every five tokens from the sequential prediction and average the metrics per steps to get the final results.

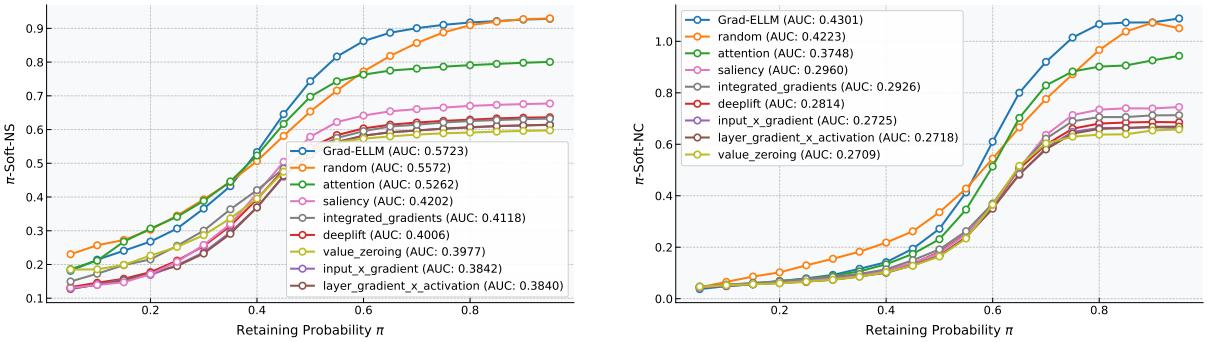
4.3 Implementation Details

Our method can select the number of layers for aggregating the approximation in (5). By default, we set the number of layers to 32 to get the most comprehensive information. The detailed prompts of different datasets are presented in Appendix part.

We use pre-trained generative LLMs from the Huggingface library (Wolf et al., 2020) with default settings. We set top_p=1 to ensure consistent output when running different attribution methods. Experiments are run on a single NVIDIA RTX3090 GPU with 24GB memory.

4.4 Quantitative Results

The AUC results for π-Soft-NS/NC are shown in Table 1. Example curves for π-Soft-NS/NC ver-



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

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

Figure 2: Proposed π-Soft-NS/NC vs. Retaining Probability on IMDb for different XAI methods with Llama.

(a) Llama: AUC of π-Soft-NS/NC (↑)

Dataset	Attn	DL	I×G	IG	LGA	Rnd	Sal	VZ	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.526	0.401	0.384	0.412	0.384	0.557	0.420	0.398	0.572
SST2	0.361	0.493	0.493	0.526	0.493	0.549	0.503	0.325	0.563
BoolQ	0.121	0.326	0.317	0.266	0.316	0.334	0.324	0.116	0.370
TellMeWhy	0.225	0.313	0.299	0.286	0.299	0.283	0.309	0.162	0.308
WikiBio	0.158	0.162	0.149	0.165	0.148	0.175	0.139	0.106	0.194
Avg	0.278	0.339	0.329	0.331	0.328	0.379	0.339	0.221	0.401
AUC π-Soft-NC ↑									
IMDb	0.375	0.281	0.273	0.293	0.272	0.422	0.296	0.271	0.430
SST2	0.165	0.255	0.258	0.276	0.258	0.318	0.253	0.136	0.304
BoolQ	0.356	1.953	1.765	1.434	1.832	2.381	1.778	0.330	2.462
TellMeWhy	0.684	1.034	0.745	1.356	0.758	0.796	0.796	0.398	1.322
WikiBio	0.571	1.007	1.829	1.003	0.845	1.475	0.660	2.396	1.059
Avg	0.430	0.906	0.974	0.872	0.793	1.078	0.756	0.706	1.115

(b) Mistral: AUC of π-Soft-NS/NC (↑)

Dataset	Attn	DL	I×G	IG	LGA	Rnd	Sal	VZ	Ours
	Methods								
AUC π-Soft-NS ↑									
IMDb	0.425	0.432	0.431	0.362	0.433	0.293	0.402	0.369	0.400
SST2	0.304	0.300	0.297	0.357	0.297	0.337	0.413	0.249	0.396
BoolQ	0.236	0.289	0.292	0.268	0.288	0.299	0.294	0.158	0.316
TellMeWhy	0.247	0.307	0.308	0.257	0.308	0.335	0.304	0.190	0.368
WikiBio	0.325	0.348	0.345	0.304	0.344	0.402	0.326	0.218	0.438
Avg	0.307	0.335	0.334	0.310	0.334	0.333	0.348	0.237	0.383
AUC π-Soft-NC ↑									
IMDb	0.346	0.415	0.422	0.357	0.421	0.618	0.388	0.065	0.530
SST2	0.200	0.278	0.277	0.366	0.275	0.560	0.347	0.099	0.464
BoolQ	0.388	0.444	0.452	0.422	0.448	0.487	0.448	0.341	0.451
TellMeWhy	0.426	0.467	0.462	0.386	0.460	0.564	0.465	0.389	0.531
WikiBio	0.410	0.391	0.392	0.334	0.389	0.512	0.376	0.351	0.477
Avg	0.354	0.399	0.401	0.373	0.399	0.548	0.405	0.249	0.491

Table 1: Faithfulness evaluation with AUC of π-Soft-NS/NC 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, LGA = Layer Gradient×Activation, Rnd = Random, Sal = Saliency, VZ = Value Zeroing.

	Attn	DL	I×G	IG	LGA	Rnd	Sal	VZ	Ours
Soft-NS↑	0.052	0.087	0.086	0.089	0.091	0.132	0.087	0.032	0.138
Soft-NC↑	0.004	0.006	0.006	0.005	0.006	0.005	0.006	0.004	0.013
$\mathbb{E}[R]$	17	20	19	18	19	80	21	8	62

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

sus retaining probability π on IMDb with Llama are presented in Figs. 2. On Llama, our method achieved the best performance according to π-Soft-NS AUC, significantly outperforming other methods. On π-Soft-NC AUC, our method also performed exceptionally well. Notably, the Value Zeroing method performed poorly on both the metrics, which is likely because its attribution scores are related to changes in predicted probability.

For Mistral, our method continues to lead in π-Soft-NS AUC, significantly outperforming other comparable methods. However, on π-Soft-NC, the Random method achieved the highest score, with our method ranking second. This may stem from Mistral's training optimizations, including sliding window attention and grouped-query attention, which foster more distributed token dependencies and uniform importance across contexts. In such models, uniform (random) perturbations can disrupt outputs more effectively for comprehensiveness metrics, as the model relies less on sparse key tokens and more on broader contextual windows, compared to Llama's more global attention patterns. Overall, our method demonstrates strong competitiveness on both metrics, especially showing stable and excellent performance on the Soft-NS metric, validating its ability to generate high-quality attributions.

The π-Soft-NS/NC metrics prioritize quality of the attribution scores as calibrated probabilities of importance, over the accuracy of the attribution orderings (as with insertion/deletion metrics). Thus, our method's superior performance on these two

metrics demonstrates its ability to produce better importance distributions.

Comparison with Soft-NC/NS: Tab. 2 shows the results when using the original Soft-NS/NC. The raw score outputs are used, and no transformations are applied. The expected number of retained words $\mathbb{E}[R]$ can be different for each method. For methods like DeepLIFT, Integrated Gradients, Input$\times$Gradient, LayerGradient$\times$Activation and Saliency, which have $\mathbb{E}[R] \approx 20$, it is unclear whether differences in Soft-NS are due to different amounts of kept information or the actual information. When evaluated this way, heat maps with larger means will inevitably yield better results because they retain more text in the soft perturbations (e.g., our method and Random). In other words, such evaluation is like erroneously comparing results with retaining probability $\pi = 0.1$ to those with $\pi = 0.4$ in Fig. 2. Therefore, simply using Soft-NS/NC on raw scores for evaluation is unfair.

4.4.1 Qualitative Results

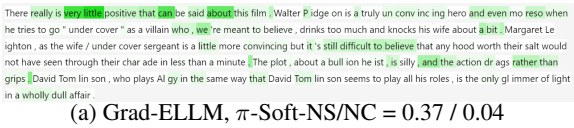
For qualitative analysis, we set the retaining probability π to the same value to fairly compare different methods. Fig. 3 and Fig. 4 shows the importance distribution given by the Grad-ELLM, Attention and Saliency on IMDb and SST2. Our method highlights the phrase "very little positive" when explaining why the model gives a negative rating, while Attention just highlights the token "positive". Saliency does not focus on the key phrase at all. On SST2, our method can even capture tokens with optimistic sentiments, such as "10/10" and ":)", in which scenario Grad-ELLM can achieve much better Soft-NS/NC. In contrast, Attention and Saliency identify the token ". :)", where the comma is not useful for prediction (as evidenced by its low π-Soft-NC). Meanwhile Saliency does not assign many weights to these key tokens, thus achieving the worst results. More qualitative results with different π are presented in Appendix.

5 Conclusion

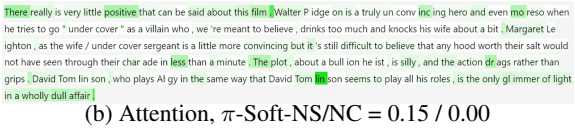
In this work, we propose a gradient-based method Grad-ELLM for calculating input attributions for decoder-only LLM. Our method generates faithful attribution heatmaps at each generation step by aggregating channel and spatial importance weights derived from gradients and attention maps, effectively highlighting the contributions of input tokens to output predictions. Through extensive experiments using open-source models like Llama and Mistral, we demonstrated that our adapted Grad-ELLM consistently outperforms baselines in terms of faithfulness metrics.

Additionally, we propose the faithfulness metrics π-Soft-NC and π-Soft-NS, which first matches the retaining probability of the attribution scores before generating the soft-perturbations for Soft-NC/NS, enabling fairer comparisons across methods with different characteristics.

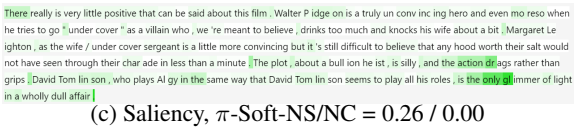
Future work could explore extensions to multimodal LLMs, integration with ethical alignment tasks like bias mitigation, and further refinements to the loosened attention mechanism for even sparser attention patterns in larger models. By enhancing transparency, our approach paves the way for more trustworthy and interpretable AI systems.



(a) Grad-ELLM, π-Soft-NS/NC = 0.37 / 0.04

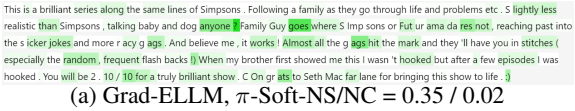


(b) Attention, π-Soft-NS/NC = 0.15 / 0.00



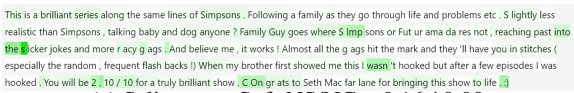
(c) Saliency, π-Soft-NS/NC = 0.26 / 0.00

Figure 3: Explanations for negative sample from IMDb. The importance of words are visualized by the degree of green color. The retaining probability is $\pi = 0.1$.



(a) Grad-ELLM, π-Soft-NS/NC = 0.35 / 0.02

(b) Attention, π-Soft-NS/NC = 0.17 / 0.00



(c) Saliency, π-Soft-NS/NC = 0.16 / 0.00

Figure 4: Explanations for positive sample from SST2. The importance of words are visualized by the degree of green color. The retaining probability is $\pi = 0.1$.

6 Limitations

Although Grad-ELLM achieves consistent superiority in faithfulness metrics across various tasks and models, it is not without limitations. Many existing attribution methods, such as Saliency and Input×Gradient, tend to produce sparse heatmaps that concentrate importance on a limited number of tokens, enabling sharp changes in model outputs during perturbation-based evaluations. In contrast, Grad-ELLM generates denser heatmaps due to the incorporation of loosened attention maps, which distribute attribution scores more broadly across input tokens. This density can make the method more susceptible to noise and accumulated errors in autoregressive generation, potentially leading to less optimal performance on insertion and deletion curves. For instance, sparse methods may achieve faster probability drops or rises by isolating highly influential tokens, whereas our denser attributions require more tokens to be perturbed to observe similar effects. Indeed our method may appear information-overloaded in scenarios requiring extremely sparse interpretation (such as identifying key diagnostic markers), but this can usually be addressed by setting a threshold to extract top-k features (or top-p retaining probability). Our core advantage lies in providing a more faithful and complete decision context.

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.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

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*.

Anonymous. 2024. Evaluating human alignment and model faithfulness of LLM rationale. In *Submitted to ACL Rolling Review - June 2024*. Under review.

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.

Reuben Binns. 2018. Fairness in machine learning: Lessons from political philosophy. In *Conference on fairness, accountability and transparency*, pages 149–159. PMLR.

Alan David Boyle, Furui Cheng, Vilém Zouhar, and Mennatallah El-Assady. 2025. Cafga: Customizing feature attributions to explain language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 461–470.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37:95764–95807.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

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.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32.

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.

Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. 2021. Layercam: Exploring hierarchical class activation maps for localization. *IEEE transactions on image processing*, 30:5875–5888.

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. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

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.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Sheng Zhang, Jin Wang, Haitao Jiang, and Rui Song. 2022. Locally aggregated feature attribution on natural language model understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2189–2201, Seattle, United States. Association for Computational Linguistics.

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*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.

A Dataset Details

More detailed descriptions of the datasets:

- **IMDb** is designed 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 designed by (Socher et al., 2013), which consist 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 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 follows the dataset setting in (Zhao and Shan, 2024; Lal et al., 2021).

- **WikiBio** is a dataset consisted of Wikipedia biographies. Following (Zhao and Shan, 2024; Manakul et al., 2023), we take the first two sentences as a prompt.

B Prompt Details

The whole prompts are shown in Tab. 3. For Llama, we add some special tokens in the end of the input text to ensure the model is output in a formatted manner. These special tokens are: "<|start_header_id|> assistant <|end_header_id|>n". These special tokens are the first tokens that Llama always outputs when it outputs content. 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.

C 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 consider each step as 5% of number of 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 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.

C.1 Results

The Insertion/Deletion AUC results are presented in Tables 4 and 5. In the experiments conducted with Llama, our method achieved the second-best result in Deletion evaluation and the third-best result in Insertion evaluation. However, our method yielded slightly worse results in the Mistral experiment, only achieving third place. 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-NS/NC metrics reveal that our method has better overall importance distributions.

D Qualitative Results

We also provide some extra qualitative results on IMDb ans SST2 with different retaining probability $\pi = \{0.25, 0.5\}$. The results are shown in Fig. 5 and Fig. 6.

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`
	`  }`
	`  ]`

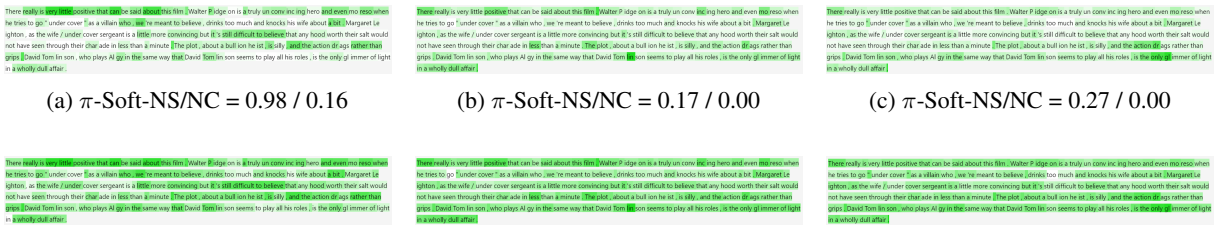
Table 3: Detailed prompts for all datasets

Deletion↑	Attention	Deeplift	Input×Gradients	Integrated Gradients	Layer Gradient×Activation	Random	Saliency	Value Zeroing	Ours (w/ | w/o)
IMDb	0.2691	0.2021	0.2031	0.1769	0.2031	0.1844	0.1736	0.2729	0.2237 | 0.2660
SST2	0.3577	0.3279	0.3291	0.3064	0.3290	0.310	0.2943	0.3755	0.3279 | 0.3639
Average	0.3134	0.2650	0.2661	0.2417	0.2660	0.2487	0.2340	0.3242	0.2878 | 0.3150
Inserton↓	Attention	Deeplift	Input×Gradients	Integrated Gradients	Layer Gradient×Activation	Random	Saliency	Value Zeroing	Ours (w/ | w/o)
IMDb	0.0917	0.1189	0.1220	0.1098	0.1221	0.1696	0.1316	0.0851	0.1204 | 0.0930
SST2	0.1584	0.1723	0.1723	0.1768	0.1722	0.2237	0.1887	0.1468	0.1891 | 0.1687
Average	0.1251	0.1456	0.1472	0.1433	0.1472	0.1967	0.1602	0.1160	0.1548 | 0.1309

Table 4: Faithfulness evaluation of Deletion and Insertion on the task of sentiment classification with Llama. The best performance, second place, and third place are marked in red, blue, and green respectively. w/o means we use the original attention rather than the loosen attention, and vice versa.

Deletion↑	Attention	Deeplift	Input×Gradients	Integrated Gradients	Layer Gradient×Activation	Random	Saliency	Value Zeroing	Ours (w/ | w/o)
IMDb	0.2853	0.2056	0.2059	0.1829	0.2060	0.1875	0.2024	0.2510	0.2452 | 0.2407
SST2	0.3853	0.3153	0.3145	0.3331	0.3146	0.3099	0.3124	0.3801	0.3485 | 0.3515
Average	0.3353	0.2605	0.2602	0.2580	0.2603	0.2487	0.2574	0.3156	0.2969 | 0.2961
Inserton↓	Attention	Deeplift	Input×Gradients	Integrated Gradients	Layer Gradient×Activation	Random	Saliency	Value Zeroing	Ours (w/ | w/o)
IMDb	0.0636	0.0888	0.0895	0.1011	0.0897	0.1614	0.1316	0.0852	0.0978 | 0.1201
SST2	0.1561	0.1791	0.1804	0.1783	0.1806	0.2287	0.1763	0.1593	0.1710 | 0.1744
Average	0.1099	0.1340	0.1350	0.1397	0.1352	0.1951	0.1308	0.1155	0.1344 | 0.1557

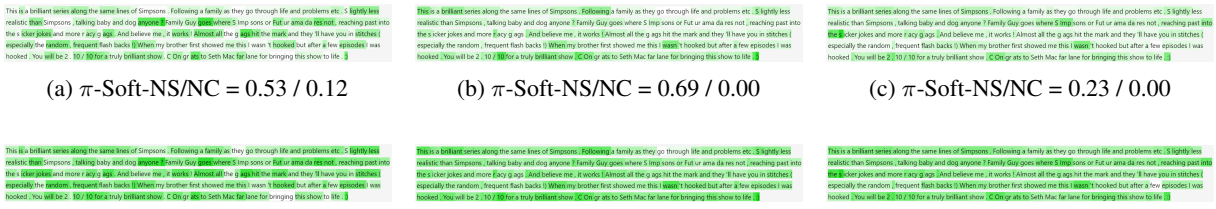
Table 5: Faithfulness evaluation of Deletion and Insertion on the task of sentiment classification with Mistral. The best performance, second place, and third place are marked in red, blue, and green respectively. w/o means we use the original attention rather than the loosen attention, and vice versa.



(a) π-Soft-NS/NC = 0.98 / 0.16 (b) π-Soft-NS/NC = 0.17 / 0.00 (c) π-Soft-NS/NC = 0.27 / 0.00

(d) π-Soft-NS/NC = 0.99 / 0.25 (e) π-Soft-NS/NC = 0.99 / 0.21 (f) π-Soft-NS/NC = 0.99 / 0.00

Figure 5: Explanations for negative sample from IMDb. The importance of words are visualized by the degree of green color. The retaining probability π of the first and second rows are $\pi = 0.25$ and $\pi = 0.5$ respectively. **Grad-ELLM**: (a) and (d); **Attention**: (b) and (e); **Saliency**: (c) and (f).



(a) π-Soft-NS/NC = 0.53 / 0.12 (b) π-Soft-NS/NC = 0.69 / 0.00 (c) π-Soft-NS/NC = 0.23 / 0.00

(d) π-Soft-NS/NC = 0.93 / 0.22 (e) π-Soft-NS/NC = 0.95 / 0.17 (f) π-Soft-NS/NC = 0.94 / 0.07

Figure 6: Explanations for positive sample from SST2. The importance of words are visualized by the degree of green color. The retaining probability π of the first and second rows are $\pi = 0.25$ and $\pi = 0.5$ respectively. **Grad-ELLM**: (a) and (d); **Attention**: (b) and (e); **Saliency**: (c) and (f).