PR-XAI: PageRank-Based Feature Attribution for Transformers

Behrooz Azarkhalili[1,2], **Linyi Li**[1], **and Maxwell Libbrecht**[1]

[1]Computing Science Department, Simon Fraser University
[2]Life Language Processing Lab, University of California, Berkeley
{bazarkha, linyi_li, maxwell_libbrecht}@sfu.ca

Abstract

We introduce PR-XAI, a feature attribution method for transformer models based on the PageRank algorithm. The proposed PR-XAI models the attention mechanism as a directed graph, with weights derived from attention weights and their gradients. Evaluations across five well-known text classification datasets and three different architectures show that PR-AG, one variant of PR-XAI, outperforms state-of-the-art attribution methods in faithfulness and classification metrics, with significant gains on long-form text.

1 Introduction

Feature attribution methods form the cornerstone of explainable AI (XAI), providing insights into model decision-making by assigning importance scores to input features. With transformers now dominant in natural language processing (Vaswani et al., 2017; Devlin et al., 2019; Sanh et al., 2020), developing effective attribution methods for these models has become critical.

Despite comprehensive research on transformer interpretability, existing feature attribution methods suffer from fundamental limitations that undermine their efficacy. Many approaches are architecturally incomplete, omitting critical components such as feed-forward networks or reducing transformers to generic neural networks. In addition, they fail to account for the mixing effect (Kobayashi et al., 2021, 2024; Modarressi et al., 2022, 2023; Ferrando et al., 2022), whereby information flows across tokens through successive attention operations, and also overlook complex pairwise feature interactions that are intrinsic to transformer architectures.

Recent studies have begun to model attention weights as graphs, leveraging graph algorithms to improve interpretability (Abnar and Zuidema, 2020; Ethayarajh and Jurafsky, 2021; Azarkhalili and Libbrecht, 2025). However, the field has yet to establish which algorithm is the most effective in this context. To address the majority of the current challenges, we propose PR-XAI, a token-level attribution method for **encoder-only** models that applies PageRank (Brin and Page, 1998) to attention-derived graphs, where edges are weighted by attention scores, their gradients, or their element-wise product. Although not previously applied in this setting, PageRank effectively measures node importance, can be computed efficiently on large networks, and has desirable theoretical properties (App. A). By utilizing recursive importance propagation, PR-XAI captures both direct and indirect token effects, yielding architecturally-aware global attributions in the form of token-level importance scores.

We have developed four key contributions:
First, we introduce the PR-XAI framework with three variants, PR-A, PR-G, and PR-AG, whose **central novelty** lies in the construction of a layered transition matrix from transformer attention and its gradient tensors (Sec. 4.2), which enables PageRank-based importance propagation across layers. This new formulation is distinct from prior graph-based methods such as Attention Rollout and Attention Flow that operate directly on attention matrices without constructing a transition matrix. The accompanying theoretical analysis in App. A guarantees the convergence and stability necessary for this formulation.

Second, a comprehensive evaluation across five datasets demonstrates that PR-AG outperforms most benchmark methods on both faithfulness and classification metrics, with statistical tests proving its effectiveness in most scenarios.

Third, we exhibit PR-AG's effectiveness on long texts and those with long-range semantic dependencies, as evidenced by superior results on the IMDB and Amazon datasets, along with our controlled contextual experiments.

Finally, we conduct a comprehensive empirical analysis that includes cross-architecture robustness evaluation and dataset-specific investigations.

2 Related Work

Transformer interpretability research encompasses various methods, each with its own strengths and limitations. Attention-based methods, such as RawAtt and Rollout (Abnar and Zuidema, 2020), directly utilize attention weights as importance scores but suffer from class-agnostic outputs and disregard all significant non-attention components of transformers (Kobayashi et al., 2020; Jain and Wallace, 2019; Serrano and Smith, 2019).

Gradient-based methods introduce more refined alternatives: Integrated Gradients (Sundararajan et al., 2017) offers theoretical guarantees yet remains model-agnostic, whereas transformer-specific variants like Grads and AttGrads (Barkan et al., 2021) combine attention weights with their gradients, albeit ReLU filtering discards potentially informative negative gradients. Advanced gradient-based methods, including CAT and AttCAT (Qiang et al., 2022), integrate encoded features and skip connections for class-specific feature attributions; however, they introduce substantial computational complexity and architectural dependencies.

Layer-wise Relevance Propagation (LRP) has been extended via PartialLRP (Voita et al., 2019) for head-level analysis and TransAtt (Chefer et al., 2021b) for full propagation; however, both methods require handling transformer-specific operations that violate standard LRP assumptions. Model-agnostic methods, such as LIME (Ribeiro et al., 2016) and KernelSHAP (Lundberg and Lee, 2017), offer theoretical guarantees but suffer from instability and high computational costs, respectively. In addition, recent token decomposition frameworks, including GlobEnc (Modarressi et al., 2022) and DecompX (Modarressi et al., 2023), enable more granular component-wise attribution but remain limited in precision due to reliance on attention rollout and constrained in scalability by substantial computational demands.

3 Preliminaries

3.1 Multi-Head Attention Mechanism

Consider an input sequence represented as $\boldsymbol{X} \in \mathbb{R}^{t \times d}$, where d denotes the dimensionality of the model's input vectors and t indicates the number of tokens. The multi-head self-attention mechanism calculates attention weights for each element in the sequence through the following steps:

- **Linear Transformation**:

$$\boldsymbol{Q}_i = \boldsymbol{X}\boldsymbol{W}_i^Q,\ \boldsymbol{K}_i = \boldsymbol{X}\boldsymbol{W}_i^K,\ \boldsymbol{V}_i = \boldsymbol{X}\boldsymbol{W}_i^V \tag{1}$$

  Here, $\boldsymbol{Q}_i, \boldsymbol{K}_i \in \mathbb{R}^{t \times d_k}$ and $\boldsymbol{V}_i \in \mathbb{R}^{t \times d_v}$, where d_k and d_v represent the dimensionality of the key vector and value vector respectively, and i represents the index of the attention head.

- **Scaled Dot-Product Attention**:

$$\boldsymbol{A}_i^*(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \widetilde{\boldsymbol{A}}_i \boldsymbol{V}_i \tag{2}$$

  where the matrix of attention weights $\widetilde{\boldsymbol{A}}_i \in \mathbb{R}^{t \times t}$ is defined as:

$$\widetilde{\boldsymbol{A}}_i = \text{softmax}\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d_k}}\right) \tag{3}$$

- **Concatenation and Linear Projection**:

$$\textbf{MultiHead}(\boldsymbol{X}) = \text{Concat}(\boldsymbol{A}_1^*, \ldots, \boldsymbol{A}_h^*)\boldsymbol{W}^O \tag{4}$$

  where the matrix $\textbf{MultiHead}(\boldsymbol{X}) \in \mathbb{R}^{t \times d}$ and the matrix $\boldsymbol{W}^O \in \mathbb{R}^{h \cdot d_v \times d}$.

For a transformer architecture comprising l attention layers, the attention weights at each layer can be characterized as multi-head attention weights:

$$\widehat{\boldsymbol{A}} = \text{Concat}(\widetilde{\boldsymbol{A}}_1, \widetilde{\boldsymbol{A}}_2, \ldots, \widetilde{\boldsymbol{A}}_h) \in \mathbb{R}^{h \times t \times t} \tag{5}$$

Extending this concept to the entire transformer architecture, the overall transformer attention weights $\boldsymbol{A}$ can be expressed as:

$$\boldsymbol{A} = \text{Concat}(\widehat{\boldsymbol{A}}_1, \widehat{\boldsymbol{A}}_2, \ldots, \widehat{\boldsymbol{A}}_l) \in \mathbb{R}^{l \times h \times t \times t} \tag{6}$$

where $\widehat{A}_j \in \mathbb{R}^{h \times t \times t}$ is the multi-head attention weight for the j-th attention layer.

3.2 PageRank Algorithm

The PageRank algorithm (Brin and Page, 1998) calculates the relative importance of nodes in a directed graph, similar to a web graph, using principles from Markov chains. It ranks nodes based on their significance through a column-stochastic matrix that represents transition probabilities between nodes.

Let $\boldsymbol{P}_{n \times n}$ be this column-stochastic transition matrix, where $\boldsymbol{P}_{ij}$ is denoting the probability of

transitioning from node j to node i. The column-stochastic property of the matrix ensures that $p_{ij} \geq 0$ and the sum of probabilities in each column equals one:

$$\sum_{i=1}^{n} p_{ij} = 1 \quad \forall j = 1, 2, \ldots, n. \tag{7}$$

To introduce the concept of teleportation, which allows random jumps between nodes, we define the teleportation parameter α, where $0 \leq \alpha < 1$. This parameter balances the probability of traversing an edge versus teleporting to a randomly selected node. The modified transition matrix $\boldsymbol{M}$ is given by:

$$\boldsymbol{M} = \alpha \boldsymbol{P} + (1 - \alpha)\boldsymbol{v}\boldsymbol{e}^T \tag{8}$$

where $\boldsymbol{e}_{n\times 1} = (1, 1, \ldots, 1)^T$ and $\boldsymbol{v}_{n\times 1}$ is the teleportation or personalization vector, satisfying $v_i \geq 0$ and $\boldsymbol{e}^T\boldsymbol{v} = 1$ for normalization. In fact, the personalization vector $\boldsymbol{v}$ in PageRank algorithm represents an initial probability distribution for a random walk on our graph,

Theorem 3.1. The matrix $\boldsymbol{M}$ defined in eq. 8 has a maximum eigenvalue of 1, associated with a unique eigenvector $\boldsymbol{x}$. Furthermore, the second largest eigenvalue of $\boldsymbol{M}$ does not exceed α (Haveliwala and Kamvar, 2003; Langville and Meyer, 2004).

Let $\boldsymbol{x}(\boldsymbol{v})$ denote the n-dimensional personalized PageRank vector associated with the corresponding personalization vector $\boldsymbol{v}$, which ranks nodes by importance. The personalized PageRank vector $\boldsymbol{x}(\boldsymbol{v})$ is derived by solving the following eigenvalue problem:

$$\begin{aligned} \boldsymbol{M}\boldsymbol{x} &= \boldsymbol{x} \\ \|\boldsymbol{x}\|_1 &= \boldsymbol{e}^T\boldsymbol{x} = 1 \end{aligned} \tag{9}$$

4 Methods

We develop the proposed method through three components: Sec. 4.1 defines information tensors aggregating attention weights and their gradients, Sec. 4.2 creates transition matrices from directed graphs with edge weights derived from information tensors, and Sec. 4.3 applies PageRank to compute token-level attribution. The theoretical explanation for the proposed method can be found in App. A.1.

4.1 Information Tensor

In transformer models, the information propagates through pathways formed by the attention mechanism. These pathways resemble routes in a graph, where tokens act as nodes and computations act as edges. The weights of these edges represent key computational quantities, which quantify the flow of information through the network (Ferrando and Voita, 2024; Mueller, 2024).

In this perspective, attention weights capture the flow of information through the neural network during the feed-forward phase, reflecting the importance of input elements in generating the output (Abnar and Zuidema, 2020; Ferrando and Voita, 2024). The gradient of attention weights, in turn, tracks how output variations affect the attention mechanism during back-propagation (Barkan et al., 2021; Chefer et al., 2021b). Utilizing both attention weights and their gradients, one can simultaneously capture information flow during feed-forward and back-propagation, offering a holistic view of the network's dynamics (Barkan et al., 2021; Qiang et al., 2022; Chefer et al., 2021b,a).

Our proposed method has been developed using the information tensor $\bar{\boldsymbol{A}} \in \mathbb{R}^{l\times t\times t}$, which is an aggregated function of the transformer attention weights $\boldsymbol{A}$, as defined in eq. 6. In this research, we introduce three distinct aggregation functions to generate the information tensors (Barkan et al., 2021; Chefer et al., 2021b,a).

1. **Attention (A):**
   $\bar{\boldsymbol{A}} := \mathbb{E}_h(\boldsymbol{A})$: This function captures the direct influence of attention mechanism in the model, similar to the methods proposed by Abnar and Zuidema (2020). By averaging across attention heads, it identifies consistent attention patterns that persist across various representational subspaces, thus representing stable token relationships.

2. **Attention Grad (G):**
   $\bar{\boldsymbol{A}} := \mathbb{E}_h(\lfloor \nabla \boldsymbol{A} \rfloor_+)$: This function isolates the sensitivity of the model's prediction to changes in attention patterns, following the gradient-based attribution principles proposed by Sundararajan et al. (2017).

3. **(Attention $\times$ Attention Grad) (AG):**
   $\bar{\boldsymbol{A}} := \mathbb{E}_h(\lfloor \boldsymbol{A} \odot \nabla \boldsymbol{A} \rfloor_+)$: This function applies a form of guided backpropagation method (Springenberg et al., 2015) especially adjusted to attention mechanism. The element-wise product between attention weights and their gradients effectively combines both forward influence and backward sensitivity, similar to

the principle behind Grad-CAM (Selvaraju et al., 2020). This function has been utilized by Chefer et al. (2021b) to develop a more accurate feature attribution by capturing both the magnitude of attention and its impact on the final prediction.

Here, $\lfloor x \rfloor_+ = \max(x, 0)$, $\odot$ denotes the Hadamard product, $\nabla \boldsymbol{A} := \frac{\partial y_t}{\partial \boldsymbol{A}}$ where y_t represents the model's scalar output for the given problem, and $\mathbb{E}_h$ denotes the mean across the heads dimension.

This formulation of information tensors bridges the gap between traditional attention-based feature attribution methods and gradient-based attribution approaches, addressing the limitations identified in previous research on transformer interpretability (Jain and Wallace, 2019; Serrano and Smith, 2019; Kobayashi et al., 2020).

4.2 Construction of Transition Matrix

This section presents our primary contribution in this work: a novel algorithm to construct a column-stochastic transition matrix $\boldsymbol{P}$ from the aggregated attention and gradient information tensor $\bar{\boldsymbol{A}}$. In contrast to previous graph-based approaches that directly aggregate attention matrices (e.g., Rollout) or compute maximum flow (e.g., Attention Flow), our method constructs a layered directed graph with explicit inter-layer edges weighted by the information tensor, enabling PageRank to propagate importance across the full transformer architecture. Specifically, we utilize the information tensor $\bar{\boldsymbol{A}}$ from Sec. 4.1 to incorporate the attention weights in deriving feature attributions by constructing a graph representation $\mathcal{G}$ of transformer or analogous attention-based models. This is accomplished by assigning edge weights in the graph $\mathcal{G}$ based on the information tensor $\bar{\boldsymbol{A}}$. The steps to construct the graph $\mathcal{G}$, including the definition of its adjacency matrix $\mathcal{A}$ and transition matrix $\boldsymbol{P}$, are outlined in Algorithm 1.

To ease the understanding of Algorithm 1, we explain how to construct the layered attribution graph $\mathcal{G}$. This graph has an adjacency matrix with dimensions $(t \cdot (l+1), t \cdot (l+1))$. We designate the nodes at layer $\ell \in \{1, \ldots, l\}$ and token $i \in \{1, \ldots, t\}$ as $v_{\ell,i}$. The following guidelines outline how to define the weight of edges between nodes:

The weight of the edge from node $v_{\ell+1,i}$ to node $v_{\ell,j}$ is defined as $\widehat{\boldsymbol{P}}[\mathrm{I}_{i,\ell+1}, \mathrm{I}_{j,\ell}] = \bar{\boldsymbol{A}}_{\ell,i,j}$, and the weight of the edge from node $v_{\ell,j}$ to node $v_{\ell+1,i}$ is defined as $\widehat{\boldsymbol{P}}[\mathrm{I}_{i,\ell}, \mathrm{I}_{j+1,\ell}] = \bar{\boldsymbol{A}}_{\ell,j,i} = \bar{\boldsymbol{A}}^T_{\ell,i,j}$ for $\ell \in \{0, \ldots, l-1\}$, $i \in \{1, \ldots, t\}$, and $j \in \{1, \ldots, t\}$, where $\mathrm{I}_{i,\ell+1} = i + t * \ell$ and $\mathrm{I}_{j,\ell} = j + t * (\ell - 1)$.

Algorithm 1 Construction of Transition Matrix

Require: $\bar{\boldsymbol{A}}_{l \times t \times t}$: An information tensor.
Ensure: $\mathcal{A}$: Adjacency Matrix, $\boldsymbol{P}$: Transition Matrix.

▷ Initialization
$Q_{tl} \leftarrow t * (l+1)$
$\widehat{\boldsymbol{P}} \leftarrow \text{zeros}(Q_{tl}, Q_{tl})$
▷ Fill j-th Layer to $(j+1)$-th Layer
for j in $0, \ldots, l-1$ **do**
  start $\leftarrow t * j$
  mid $\leftarrow t * (j+1)$
  end $\leftarrow t * (j+2)$
  $\widehat{\boldsymbol{P}}$[start:mid , mid:end] $\leftarrow \bar{\boldsymbol{A}}_{[j,:,:]}$
  $\widehat{\boldsymbol{P}}$[mid:end , start:mid] $\leftarrow \bar{\boldsymbol{A}}^T_{[j,:,:]}$
end for
▷ Compute the sum along the last axis
$\widehat{\boldsymbol{P}}_{\text{sum}} \leftarrow \widehat{\boldsymbol{P}}.\text{sum}(\text{dim} = -1,\ \text{keepdim=True})$
▷ Normalize by dividing by the sum
$\boldsymbol{P} \leftarrow \frac{\widehat{\boldsymbol{P}}}{\widehat{\boldsymbol{P}}_{\text{sum}}}$
$\mathcal{A} \leftarrow \mathbb{I}_{(\boldsymbol{P}>0)}$

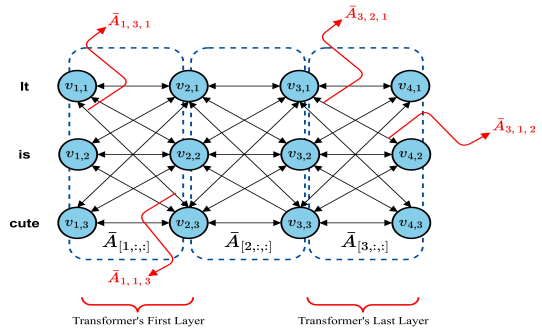
Fig. 1a depicts the schematic graph constructed from the information tensor $\bar{\boldsymbol{A}} \in \mathbb{R}^{3\times3\times3}$ utilizing Algorithm 1. Fig. 1b presents the corresponding adjacency and transition matrix derived from the information tensor $\bar{\boldsymbol{A}} := \mathbb{E}_h(\lfloor \boldsymbol{A} \odot \nabla \boldsymbol{A} \rfloor_+)$ associated with the tokens {[CLS], it, is, cute, ., [SEP]} from the sentence "It is cute." where we use the transformer model "distilbert-base-uncased". Given that the above sentence comprises 6 tokens and the transformer model includes 6 layers, each dimension of the transition matrix is $6 \times 7 = 42$.

Finally, we solve the PageRank problem defined on this graph and normalize the score to determine the importance of the input features or tokens.

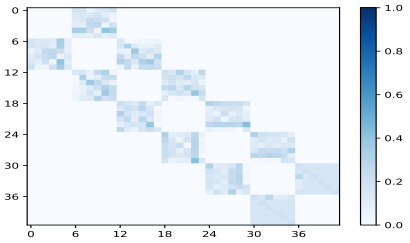
4.3 Calculating PageRank Values

In the eigenvalue problem $\boldsymbol{C}\boldsymbol{x} = \lambda \boldsymbol{x}$, the power method is a well-established algorithm to compute the eigenvector associated with the dominant eigenvalue, which is defined as the largest eigenvalue in terms of magnitude. Initiating from an arbitrary vector $\boldsymbol{x}^{(0)}$, the iterative procedure described in eq. 10 converges to the eigenvector associated with the dominant eigenvalue, provided that the largest eigenvalue of $\boldsymbol{C}$ is both real and unique.

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \rho^{(k+1)} \boldsymbol{C}\boldsymbol{x}^{(k)} \\ \rho^{(k+1)} &= \|\boldsymbol{C}\boldsymbol{x}^{(k)}\|_1^{-1} \end{aligned} \tag{10}$$



(a) Graph of information flow created via Algorithm 1.



(b) Transition matrix created via Algorithm 1.

Figure 1: Schematic representation of the graph construction and the corresponding transition matrix for the sentence "It is cute." using DistilBERT, derived from the information tensor $\bar{\boldsymbol{A}} := \mathbb{E}_h(\lfloor \boldsymbol{A} \odot \nabla \boldsymbol{A} \rfloor_+)$ using Algorithm 1.

Subsequently, we apply the power method to the PageRank eigensystem defined in eq. 9. Since the dominant eigenvalue of $\boldsymbol{M}$ is 1, normalization at each iterative step is unnecessary. By omitting the normalization process and expanding $\boldsymbol{M}$, we obtain the following iterative procedure:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{M}\boldsymbol{x}^{(k)} = \alpha \boldsymbol{P}\boldsymbol{x}^{(k)} + (1-\alpha)\boldsymbol{v}\boldsymbol{e}^T\boldsymbol{x}^{(k)} \tag{11}$$

Recalling that $\boldsymbol{e}^T\boldsymbol{x} = 1$, a simple calculation shows that if $\boldsymbol{e}^T\boldsymbol{x}^{(0)} = 1$, then $\boldsymbol{e}^T\boldsymbol{x}^{(k)} = 1$ for all subsequent iterations, leading to:

$$\boldsymbol{x}^{(k+1)} = \alpha \boldsymbol{P}\boldsymbol{x}^{(k)} + (1-\alpha)\boldsymbol{v} \tag{12}$$

Theorem 4.1. Assuming $\boldsymbol{x}$ is the exact PageRank vector that satisfies eq. 12, the following inequality holds when applying the power method to compute PageRank.

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}\|_1 \leq \alpha \|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_1 \tag{13}$$

Hence, the power method has a linear convergence rate for the PageRank problem, characterized by a convergence rate of α (Langville and Meyer, 2004; Haveliwala and Kamvar, 2003).

Theorem 4.2. Let $\boldsymbol{P}_{n\times n}$ be a column-stochastic transition matrix with $\boldsymbol{P}_{ii} = 0$ for all $1 \leq i \leq n$. The condition number for the PageRank problem defined in eq. 9 is given by $\kappa_1 = \frac{1+\alpha}{1-\alpha}$ (Kamvar and Haveliwala, 2003; Langville and Meyer, 2004).

App. A provides a detailed discussion of why our method is effective in practice, including the stability, sensitivity, and convergence analysis of the PageRank algorithm using Theorem 4.1 and Theorem 4.2. Following the standard PageRank convention (Hagberg et al., 2008), we similarly set $\alpha = 0.85$ across all experiments, which ensures computational stability by preserving a sufficient spectral gap (App. A.4).

The overall time complexity of our proposed method is $O(nl^2t^2)$, where n is the number of PageRank iterations, l is the number of transformer layers, and t is the input sequence length. While it is higher than simpler attention-based methods, the runtime remains competitive with other gradient-based methods (App. B.3), and the superior faithfulness, especially on long-form text, justifies the cost.

5 Experiments

In this section, we present a detailed evaluation of the effectiveness of our methods for sequence classification. Although our approach is versatile and can be applied to various NLP tasks, such as question answering and named entity recognition, which use **encoder-only** transformer architectures, this evaluation is specifically centered on sequence classification. All the additional details regarding our experiments can be found in App. B.

5.1 Models & Datasets

In our evaluations, we use a specific pre-trained model from the Hugging Face (Wolf et al., 2020) for each dataset and compare the performance of our explanation methods against others to assess their performance. All pretrained models listed in Tab. 5 have a total of 109, 483, 778 parameters and consist of 12 transformer blocks.

The evaluation of our proposed method includes binary classification tasks across several datasets, including SST2 (Socher et al., 2013), Yelp Polarity (Zhang et al., 2016), Amazon Polarity (McAuley and Leskovec, 2013), IMDB (Maas et al., 2011) along with multi-class classification on the AG News dataset (Zhang et al., 2015). To scale down computational costs, we conduct the experiments using a randomly selected subset of 5,000 samples from the Amazon, Yelp, and IMDB test datasets,

while using the complete test sets for the remaining datasets.

5.2 Benchmark Methods

Our experiment compares three methods defined based on the information tensors in Sec. 4.1, which we refer to as PR-A, PR-G, and PR-AG, against various baseline explanation methods designed for transformer models. To implement attention-based methods, such as RawAtt and Rollout (Abnar and Zuidema, 2020), gradient-based methods including Grads and AttGrads (Barkan et al., 2021), CAT, and AttCAT (Qiang et al., 2022), as well as LRP-based methods like PartialLRP (Voita et al., 2019) and TransAtt (Chefer et al., 2021b), we leverage the repository developed by Qiang et al. (2022). We also implement GlobEnc (Modarressi et al., 2022) and DecompX (Modarressi et al., 2023) methods using their repositories. Furthermore, we implement classical attribution methods, including KernelShap (Lundberg and Lee, 2017), Integrated Gradient (Sundararajan et al., 2017), and LIME (Ribeiro et al., 2016), using the Captum package (Kokhlikyan et al., 2020).

5.3 Evaluation Metric

AOPC: One of the principal evaluation metrics utilized in our experiments is the Area Over the Perturbation Curve (AOPC), which measures the impact of masking top $k\%$ tokens on the average change in predicted probability across all examples. The AOPC is calculated as follows:

$$\mathrm{AOPC}(k) = \frac{1}{N}\sum_{i=1}^{N} p\left(\hat{y}|\boldsymbol{x}_i\right) - p\left(\hat{y}|\tilde{\boldsymbol{x}}_i^k\right) \quad (14)$$

where N is the number of examples, $\hat{y}$ is the predicted label, $p(\hat{y}|\cdot)$ is the probability on the predicted label, and $\tilde{\boldsymbol{x}}_i^k$ is constructed by masking the $k\%$ top-scored tokens from $\boldsymbol{x}_i$. To avoid arbitrary choices for k, we systematically mask $10\%, 20\%, \ldots, 90\%$ of the tokens in order of decreasing saliency, resulting in $\tilde{\boldsymbol{x}}_i^{10}, \tilde{\boldsymbol{x}}_i^{20}, \ldots, \tilde{\boldsymbol{x}}_i^{90}$. Higher values of AOPC are considered favorable, indicating a greater influence and importance of the masked tokens on the model's output.

LOdds: Log-odds is calculated by averaging the difference of negative logarithmic probabilities on the predicted label over all test examples before and after masking $k\%$ top-scored tokens.

$$\mathrm{LOdds}(k) = \frac{1}{N}\sum_{i=1}^{N} \log \frac{p\left(\hat{y}|\tilde{\boldsymbol{x}}_i^k\right)}{p\left(\hat{y}|\boldsymbol{x}_i\right)} \quad (15)$$

6 Results

6.1 Main Metrics

We assessed the efficacy of the methods by masking the most important $k\%$ of tokens across multiple datasets, and subsequently analyzing their AOPC and LOdds scores, as illustrated in Tab. 1. Here, k denotes the percentage of tokens masked based on their attribution scores, and we systematically vary $k \in \{10, 20, \ldots, 90\}$. The values reported in Tab. 1 and Tab. 2 represent the mean AOPC and LOdds scores averaged across all k values. We employ `[MASK]` token replacement to effectively mask the selected tokens. Next, we evaluated the effectiveness of the methods by masking the least important $k\%$ of tokens across the same datasets, with the results displayed in Tab. 2.

Examining Tab. 1 and Tab. 2, it is evident that PR-AG exhibits remarkable consistency, achieving the highest-level performance across most datasets in both the top $k\%$ and bottom $k\%$ token masking scenarios. This notable consistency is especially clear in the SST2, IMDB, Amazon, and AG News datasets, where PR-AG excels in both the AOPC and LOdds metrics.

However, the Yelp dataset presents an interesting exception, where PartialLRP achieves the highest AOPC for top token masking and AttCAT performs the best for bottom token masking. We extensively explore the challenges posed by the Yelp dataset regarding our proposed method in App. C.6.

To further evaluate the performance of our proposed methods, we employ a complementary classification-based strategy. To achieve this, the top-$k\%$ most important tokens identified by each method are masked, and the model's performance is re-evaluated. We report the mean of four main classification metrics (F1, accuracy, precision, and recall) evaluated on the perturbed data in Tab. 7, with lower scores reflecting stronger performance.

We also investigated the effectiveness of these methods in the previously mentioned scenarios leveraging various encoder-only architectures, such as BERT, DistilBERT, and RoBERTa. A detailed discussion of our findings can be found in App. C.4.

Methods	SST2		IMDB		Yelp		Amazon		AG News	
	AOPC↑	LOdds↓	AOPC↑	LOdds↓	AOPC↑	LOdds↓	AOPC↑	LOdds↓	AOPC↑	LOdds↓
RawAtt	0.348	-0.973	0.329	-1.393	0.383	-1.985	0.353	-1.593	0.301	-1.105
Rollout	0.322	-0.887	0.354	-1.456	0.260	-0.987	0.304	-1.326	0.249	-0.983
Grads	0.354	-0.313	0.324	-1.271	0.412	-1.994	0.405	-1.793	0.327	-1.319
AttGrads	0.367	-0.654	0.337	-1.226	0.423	-1.978	0.419	-1.918	0.348	-1.477
CAT	0.369	-1.175	0.332	-1.274	0.417	-1.992	0.381	-1.639	0.325	-1.226
AttCAT	0.405	-1.402	0.371	-1.642	0.431	**-2.134**	0.427	-2.041	0.387	-1.688
PartialLRP	0.371	-1.171	0.323	-1.321	**0.443**	-2.018	0.384	-1.945	0.356	-1.627
TransAtt	0.399	-1.286	0.355	-1.513	0.411	-1.473	0.375	-1.875	0.377	-1.318
LIME	0.362	-1.056	0.347	-1.379	0.361	-1.568	0.358	-1.612	0.349	-1.538
KernelShap	0.382	-1.259	0.367	-1.423	0.385	-1.736	0.374	-1.717	0.351	-1.413
IG	0.401	-1.205	0.350	-1.443	0.409	-1.924	0.434	-2.024	0.393	-1.681
DecompX	0.396	-1.115	0.343	-1.411	0.401	-1.887	0.391	-1.912	0.396	-1.385
GlobEnc	0.373	-1.095	0.330	-1.323	0.388	-1.826	0.367	-1.861	0.373	-1.496
PR-A	0.372	-0.991	0.361	-1.428	0.394	-1.965	0.387	-1.761	0.381	-1.473
PR-G	0.365	-0.943	0.355	-1.247	0.406	-1.997	0.413	-2.023	0.372	-1.415
PR-AG	**0.412***	**-1.423***	**0.380***	**-1.725***	0.437	-2.027	**0.443***	**-2.110***	**0.402***	**-1.712***

Table 1: Mean AOPC and LOdds scores, averaged across $k \in \{10, 20, \ldots, 90\}$ masking steps, are reported for all methods across datasets when masking the **top** $k\%$ tokens. Higher AOPC and lower LOdds indicate stronger identification of important tokens. Best results are in bold. * shows statistically significant improvements of PR-AG over the best baseline under the ASO test ($\epsilon_{\min} \leq 0.5$; App. C.3).

Methods	SST2		IMDB		Yelp		Amazon		AG News	
	AOPC↓	LOdds↑	AOPC↓	LOdds↑	AOPC↓	LOdds↑	AOPC↓	LOdds↑	AOPC↓	LOdds↑
RawAtt	0.184	-0.693	0.151	-0.471	0.157	-0.747	0.129	-0.281	0.101	-0.427
Rollout	0.221	-0.773	0.123	-0.425	0.169	-0.734	0.171	-0.368	0.117	-0.471
Grads	0.234	-0.776	0.083	-0.203	0.131	-0.641	0.134	-0.254	0.083	-0.390
AttGrads	0.217	-0.713	0.088	-0.243	0.127	-0.603	0.135	-0.266	0.071	-0.351
CAT	0.247	-0.874	0.099	-0.327	0.134	-0.659	0.126	-0.240	0.104	-0.419
AttCAT	0.143	-0.412	0.041	-0.092	**0.103**	**-0.339**	0.115	-0.148	0.057	-0.219
PartialLRP	0.163	-0.527	0.057	-0.116	0.116	-0.486	0.167	-0.327	0.056	-0.204
TransAtt	0.148	-0.483	0.045	-0.107	0.123	-0.538	0.113	-0.140	0.049	-0.173
LIME	0.173	-0.603	0.076	-0.141	0.143	-0.687	0.158	-0.263	0.075	-0.372
KernelShap	0.197	-0.729	0.039	-0.084	0.135	-0.645	0.174	-0.351	0.067	-0.219
IG	0.150	-0.532	0.026	-0.064	0.130	-0.617	0.134	-0.241	0.052	-0.191
DecompX	0.161	-0.578	0.077	-0.134	0.121	-0.557	0.141	-0.227	0.073	-0.311
GlobEnc	0.175	-0.592	0.086	-0.161	0.139	-0.638	0.152	-0.253	0.078	-0.345
PR-A	0.191	-0.751	0.061	-0.124	0.126	-0.521	0.129	-0.253	0.066	-0.231
PR-G	0.154	-0.517	0.041	-0.095	0.118	-0.417	0.139	-0.235	0.073	-0.252
PR-AG	**0.122***	**-0.351***	**0.012***	**-0.041***	0.107	-0.394	**0.104***	**-0.123***	**0.043***	**-0.151***

Table 2: Mean AOPC and LOdds scores, averaged across $k \in \{10, 20, \ldots, 90\}$ masking steps, are reported for all methods across datasets when masking the **bottom** $k\%$ tokens. Lower AOPC and higher LOdds indicate stronger distinction between important and unimportant tokens. Best results are in bold. * shows statistically significant improvements of PR-AG over the best baseline under the ASO test ($\epsilon_{\min} \leq 0.5$; App. C.3).

6.2 Gradient-Based vs. Attention-Based vs. Hybrid Methods

A clear performance hierarchy is evident across all datasets. Pure attention-based methods, such as RawAtt and Rollout, consistently underperform compared to gradient-based approaches like Grads, AttGrads, and CAT. Furthermore, methods that fuse both attention and gradient information, such as AttCAT and PR-AG, outperform these gradient-based techniques. This pattern suggests that raw attention weights alone do not provide sufficient explanatory signals. This limitation most likely arises from their inability to capture how attention patterns influence the final predictions.

The most noticeable insight is how hybrid feature attribution methods can utilize complementary information sources to create more comprehensive attributions. AttCAT consistently outperforms both CAT and AttGrads across all datasets, just as PR-AG surpasses both PR-A and PR-G. This consistent trend indicates that combining forward attention patterns, which represent what the model focuses on, with backward gradient signals, which affects predictions, develop a synergistic representation of feature attribution that aligns more closely with the actual decision-making processes of the model.

6.3 Short-form vs. Long-form Text

PR-AG demonstrates exceptional performance on both SST2 (short movie reviews) and IMDB (long movie reviews), outperforming other competing methods. In contrast, methods like RawAtt and Rollout show significantly lower performance on longer input sequences, highlighting their limited

Method	AOPC ↑						LOdds ↓					
	Q_{25}	Q_{50}	Q_{75}	Q_{100}	**Mean ↑**	**Std ↓**	Q_{25}	Q_{50}	Q_{75}	Q_{100}	**Mean ↓**	**Std ↓**
PR-AG	**0.417***	**0.406***	**0.390***	**0.378***	**0.398***	**0.015**	**-2.015***	**-1.892***	**-1.671***	**-1.441***	**-1.755***	**0.219**
AttCAT	0.406	0.388	0.362	0.347	0.376	0.023	-2.017	-1.973	-1.561	-1.276	-1.707	0.306
RawAtt	0.335	0.317	0.304	0.283	0.310	0.019	-1.536	-1.278	-1.112	-0.921	-1.212	0.225
KernelShap	0.412	0.397	0.377	0.358	0.386	0.020	-1.975	-1.784	-1.455	-1.237	-1.612	0.286

Table 3: IMDB AOPC (↑) and LOdds (↓) scores are reported per method with mean and standard deviation, stratified by sequence length quartiles (Q_{25}, Q_{50}, Q_{75}, Q_{100}), where Q_p denotes the subset of samples with sequence lengths up to the p-th percentile of the maximum sequence length. Higher AOPC and lower LOdds indicate superior performance, with best results in bold. PR-AG achieves the highest mean AOPC, lowest mean LOdds, and minimal variance across text lengths, demonstrating robustness for longer sequences. * shows statistically significant under the ASO test ($\epsilon_{\min} \leq 0.5$; App. C.3).

capability to model longer texts, which PR-AG addresses through graph-based propagation.

To assess the relationship between sequence length and explanation quality, we also conducted stratified analysis on the IMDB dataset, splitting samples into quartiles at 25%, 50%, 75%, and 100% of maximum sequence length while preserving label distribution balance. Using ModernBERT (Warner et al., 2024), which supports sequences up to 8192 tokens, we assessed AOPC and LOdds metrics on 2,000 samples to guarantee statistical reliability. We note that ModernBERT was selected for its architectural improvements that yield better performance on longer sequences within the IMDB distribution, rather than for processing sequences at its maximum capacity of $8k$. We examined how PR-AG's performance advantage changes with increasing text length. We compared PR-AG against its top competitors, AttCAT and KernelShap, alongside RawAtt as the direct attention aggregation without propagation.

As illustrated in Tab. 3, PR-AG outperforms other methods across all quartiles, reaching the highest mean AOPC with minimal variance. KernelShap ranks second in mean AOPC with higher variance, while AttCAT ranks third in mean AOPC but has the highest variance among all methods. RawAtt undergoes significant degradation, getting the lowest mean AOPC and moderate variance. For LOdds metric, PR-AG has the most negative mean scores, indicating the highest faithfulness, followed by KernelShap and AttCAT, with RawAtt showing the weakest performance. It is noteworthy that PR-AG's advantage becomes more pronounced when the length of input sequences increases, supporting the claim that PR-AG is especially effective for modeling longer input sequences. Moreover, PR-AG's effectiveness across varying input sequence lengths also demonstrates that its PageRank-based approach can effectively identify important tokens regardless of input sequence length.

6.4 Long-Range Semantic Dependencies

The performance gap between PR-AG and simpler competing methods is most pronounced on datasets with long-range semantic dependencies, such as IMDB and Amazon. Conversely, the difference is less pronounced on datasets with more localized semantic dependencies, like AG News. This pattern indicates that PR-AG's main advantage lies in its capability to model complex dependencies between tokens utilizing PageRank's recursive importance propagation. This capability is highly valuable for texts that state sentiment through subtle linguistic constructions.

To rigorously evaluate PR-AG's capacity for modeling long-range semantic dependencies, we developed a controlled experimental framework that systematically extended input contexts while preserving semantic integrity. In this experiment, we randomly selected 5,000 reviews from IMDB dataset and 1,000 sentiment-neutral sentences from the OpenWebText Corpus (Gokaslan and Cohen, 2019) to construct a composite structure: [prefix] + [review] + [suffix], where for each review, suffix and prefix were chosen randomly from sentiment-neutral sentences. The candidate sentences were subjected to two filtering steps: (i) selecting those labeled as neutral by a pre-trained sentiment classifier (Loureiro et al., 2022), and (ii) keeping those sentences with a [CLS] vector cosine similarity to the original review below 0.1. In this framework, the distance from the leading [CLS] token to the review serves as a controlled proxy for long-range semantic dependency and the model's capability to preserve semantic focus across extended contexts. We developed three experimental conditions: **Baseline** (using original text), **Medium Context** (using suffix and prefix with 50-100 tokens), and

Extended Context (using suffix and prefix with 100-200 tokens). Classification metrics are above 92% across all configurations, confirming that the insertion of neutral content preserved task integrity.

The robustness of the methods was evaluated using the Original Text Focus (OTF) metric, which measures the proportion of the top-$k\%$ important tokens in the original sequence and the corresponding extended one, averaged across all sequences. While the **Baseline** condition naturally results in OTF $= 1.0$, we compared PR-AG to its main competitors, AttCAT and KernelShap, along with RawAtt as a baseline across multiple experimental runs.

As evidenced in Tab. 4, PR-AG exhibits robust performance across context extensions, achieving a superior OTF score in **Medium** and **Extended** contexts, consistently outperforming most methods. However, RawAtt consistently records the lowest OTF scores in most conditions. This confirms the limitations of attention-based models and validates PR-AG's graph-based propagation as superior for modeling long-range semantic dependencies.

Method	**OTF** ($k = 10\%$)			**OTF** ($k = 25\%$)		
	Baseline	**Medium**	**Extended**	**Baseline**	**Medium**	**Extended**
PR-AG	1.00	**0.78**	**0.71**	1.00	**0.85**	**0.74**
AttCAT	1.00	0.67	0.68	1.00	0.80	0.71
RawAtt	1.00	0.67	0.61	1.00	0.61	0.56
KernelShap	1.00	0.73	0.67	1.00	0.76	0.69

Table 4: OTF score under Baseline, Medium, and Extended contexts for $k = 10\%$ and $k = 25\%$. Higher values indicate better retention of important tokens from the original review segment, and the best results are in bold. PR-AG retains focus on original content better than competitors, demonstrating superior handling of long-range semantic dependencies.

7 Conclusion

This paper proposes PR-XAI, an efficient token-level feature attribution method that integrates the PageRank algorithm with the attention mechanism. Combining attention weights with their gradients, PR-AG, one variant of PR-XAI, consistently outperforms the existing benchmark methods across multiple text classification datasets and remains robust across BERT, DistilBERT, and RoBERTa, regardless of model size or pretraining strategy. These findings establish our proposed PR-XAI as a principled and scalable method for transformer interpretability.

Limitations

A key limitation of our proposed method is its current inapplicability to decoder-only architectures such as the GPT series. This restriction stems not from the methodology itself but from a structural incompatibility between PageRank and the causal nature of autoregressive attention.

Our proposed method computes the importance of tokens by modeling the information flow as a recursive random walk on graphs constructed from attention weights and their gradients, converging to a stationary distribution of importance. In decoder-only models, however, causal attention enforces unidirectional flow, yielding a directed graph with an upper-triangular adjacency matrix. Information flow can propagate only forward in this graph, from early to late tokens, generating a highly skewed importance distribution in which later tokens will accumulate disproportionate importance.

In future research, we will investigate various importance distribution correction techniques to enhance the capabilities of PR-XAI, allowing it to effectively support decoder-only models.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. *Preprint*, arXiv:2005.00928.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. *Preprint*, arXiv:1907.10902.

Behrooz Azarkhalili and Maxwell W. Libbrecht. 2025. Generalized Attention Flow: Feature Attribution for Transformer Models via Maximum Flow. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19954–19974, Vienna, Austria. Association for Computational Linguistics.

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, Virtual Event Queensland Australia. ACM.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117.

Hila Chefer, Shir Gur, and Lior Wolf. 2021a. Generic Attention-model Explainability for Interpreting Bi-

Modal and Encoder-Decoder Transformers. *Preprint*, arXiv:2103.15679.

Hila Chefer, Shir Gur, and Lior Wolf. 2021b. Transformer Interpretability Beyond Attention Visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, Nashville, TN, USA. IEEE.

Grace E. Cho and Carl D. Meyer. 2000. Markov chain sensitivity measured by mean first passage times. *Linear Algebra and its Applications*, 316(1):21–28.

E. del Barrio, J. A. Cuesta-Albertos, and C. Matrán. 2017. An optimal transportation approach for assessing almost stochastic order. *Preprint*, arXiv:1705.01788.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Preprint*, arXiv:1810.04805.

Kawin Ethayarajh and Dan Jurafsky. 2021. Attention Flows are Shapley Value Explanations. *Preprint*, arXiv:2105.14652.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the Mixing of Contextual Information in the Transformer.

Javier Ferrando and Elena Voita. 2024. Information Flow Routes: Automatically Interpreting Language Models at Scale. *Preprint*, arXiv:2403.00824.

Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. A Large-Scale Study of the Evolution of Web Pages.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Gene H. Golub and Carl D. Meyer, Jr. 1986. Using the QR Factorization and Group Inversion to Compute, Differentiate, and Estimate the Sensitivity of Stationary Probabilities for Markov Chains. *SIAM Journal on Algebraic Discrete Methods*, 7(2):273–281.

Gene H. Golub and Charles F. Van Loan. 2013. *Matrix Computations*, fourth edition edition. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Python in Science Conference*, pages 11–15, Pasadena, California.

Taher H. Haveliwala and S. Kamvar. 2003. The Second Eigenvalue of the Google Matrix.

Ilse C. F. Ipsen and Rebecca S. Wills. 2008. Mathematical properties and analysis of Google's PageRank.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepandar Kamvar and Taher Haveliwala. 2003. The Condition Number of the PageRank Problem. http://ilpubs.stanford.edu:8090/597/.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2024. Analyzing Feed-Forward Blocks in Transformers through the Lens of Attention Maps. *Preprint*, arXiv:2302.00456.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. *Preprint*, arXiv:2009.07896.

Amy Langville and Carl Meyer. 2004. Deeper Inside PageRank. *Internet Mathematics*, 1(3):335–380.

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.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. TimeLMs: Diachronic Language Models from Twitter. *Preprint*, arXiv:2202.03829.

Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. *Preprint*, arXiv:1705.07874.

Andrew L. 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, Portland, Oregon, USA. Association for Computational Linguistics.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172, Hong Kong China. ACM.

Carl D. Meyer. 1993. The Character of a Finite Markov Chain. In *Linear Algebra, Markov Chains, and Queueing Models*, pages 47–58, New York, NY. Springer.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. DecompX: Explaining Transformers Decisions by Propagating Token Decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2649–2664, Toronto, Canada. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying Global Token Attribution by Incorporating the Whole Encoder Layer in Transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Aaron Mueller. 2024. Missed Causes and Ambiguous Effects: Counterfactuals Pose Challenges for Interpreting Neural Networks. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. 2022. AttCAT: Explaining Transformers via Attentive Class Activation Tokens. In *Advances in Neural Information Processing Systems*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Preprint*, arXiv:1602.04938.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2):336–359.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. *Preprint*, arXiv:1412.6806.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. *Preprint*, arXiv:1703.01365.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. Deep-significance - Easy and Meaningful Statistical Significance Testing in the Age of Neural Networks. *Preprint*, arXiv:2204.06815.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. *Preprint*, arXiv:2412.13663.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *Preprint*, arXiv:1910.03771.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level Convolutional Networks for Text Classification. *Preprint*, arXiv:1509.01626.

A Methods

A.1 Intuition Behind PageRank

PageRank is a natural fit for our feature attribution problem since it is explicitly designed to calculate the stationary distribution of a random walk on a graph.

In our proposed method, this random walk is a concrete process defined on the multi-layered graph constructed in Algorithm 1. In this graph, a "random walker" begins at an arbitrary node (a token at a specific layer) and, at each step, moves to a subsequent node (a token in the next layer) with a probability proportional to the weight of the edge connecting them in the transition matrix $\boldsymbol{P}$. These edge weights are derived directly from the information tensor $\bar{\boldsymbol{A}}$, meaning the walker is more likely to traverse pathways with high attention flow and high gradient sensitivity.

This procedure models the stepwise propagation of information across layers. The PageRank score of a token thus reflects the long-term probability of the walker residing on that node, serving as a direct measure of its global importance in the network.

By aggregating influence over paths of all lengths, the method captures both direct attention connections and indirect, multi-hop dependencies. This provides a key advantage over methods limited to direct attention scores (e.g., RawAtt) or raw gradient magnitudes, which fail to account for the cumulative effects of token interactions.

Our results exhibited in Sec. 6.3 and Sec. 6.4, where PR-AG outperforms other methods on long-form texts, empirically corroborate this theoretical justification, as long-range semantic dependencies are more prevalent in longer sequences.

A.2 Stability of PageRank

The most significant consequence of Theorem 4.1 pertains to the stability of PageRank. Define the perturbed matrix as $\widetilde{\boldsymbol{M}} = \boldsymbol{M} + \epsilon \boldsymbol{B}$, where $\epsilon \boldsymbol{B}$ is the "error matrix" representing the variations in the matrix $\boldsymbol{M}$. Let $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ denote the PageRank vectors associated with the matrices $\boldsymbol{M}$ and $\widetilde{\boldsymbol{M}}$, respectively. It is known that for the system of linear equations described in eq. 9, we have:

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1 \leq \kappa_1 \epsilon \|\boldsymbol{B}\|_1 \tag{16}$$

From Theorem 4.2, we can rewrite this as:

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1 \leq \epsilon \frac{1+\alpha}{1-\alpha} \|\boldsymbol{B}\|_1 \tag{17}$$

Here, ϵ is a theoretical parameter representing the magnitude of a small perturbation to the initial state x_0, whereas α is a hyperparameter of the PageRank algorithm set by the user.

This implies that, for values of α close to 1, the PageRank algorithm exhibits instability, where a minor perturbation in $\boldsymbol{M}$ can lead to a significant variation in PageRank vector. Conversely, for lower values $0.8 \leq \alpha \leq 0.9$, the algorithm demonstrates stability, ensuring that small changes in $\boldsymbol{M}$ result in only minor fluctuations in the final PageRank vector (Kamvar and Haveliwala, 2003; Langville and Meyer, 2004).

A.3 Sensitivity of PageRank

In this section, we present a rigorous analysis of PageRank's sensitivity to various perturbations.

Theorem A.1 (PageRank Sensitivity). Let M be the transition matrix defined in eq. 8. Then the following sensitivity bounds hold (Ipsen and Wills, 2008):

(i) **Graph Structure Sensitivity:** If $\tilde{P}$ is another column-stochastic matrix, and $\tilde{\boldsymbol{x}}$ is the PageRank vector of $\widetilde{M} = \alpha \tilde{P} + (1-\alpha) v e^T$, then

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq \frac{\alpha}{1-\alpha} \|\tilde{P} - P\|_1 \tag{18}$$

(ii) **Teleportation Sensitivity:** If $\hat{\alpha}$ is a new teleportation parameter with $0 \leq \hat{\alpha} < 1$, and $\tilde{\boldsymbol{x}}$ is the PageRank vector of $\widetilde{M} = \hat{\alpha} P + (1-\hat{\alpha}) v e^T$, then

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq \frac{2}{1-\alpha} |\alpha - \hat{\alpha}| \tag{19}$$

(iii) **Personalization Sensitivity:** If $\hat{v}$ is another personalization vector with $\|\hat{v}\|_1 = 1$, and $\tilde{\boldsymbol{x}}$ is the PageRank vector of $\widetilde{M} = \alpha P + (1-\alpha) \hat{v} e^T$, then

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq \|v - \hat{v}\|_1 \tag{20}$$

When discussing sensitivity and stability within the PageRank framework, it is important to analyze how perturbations in the matrix $\boldsymbol{P}$ affect the vector $\boldsymbol{x}$. This analysis requires distinguishing between two mathematical formulations: the linear system $(\boldsymbol{I} - \alpha \boldsymbol{P})\boldsymbol{x} = (1-\alpha)\boldsymbol{v}$ and the eigenvector problem $(\alpha \boldsymbol{P} + (1-\alpha)\boldsymbol{v}\boldsymbol{e}^T)\boldsymbol{x} = \boldsymbol{x}$.

The PageRank condition number $\kappa_1 = \frac{1+\alpha}{1-\alpha}$ (Kamvar and Haveliwala, 2003) demonstrates that as $\alpha \to 1$, the linear system becomes increasingly ill-conditioned, amplifying the perturbations of the solution vector under minor matrix perturbations.

nevertheless, this numerical instability does not inherently propagate to the eigenvector formulation.

Although the entries of the solution vector may vary significantly under matrix perturbations, the directional stability of $\boldsymbol{x}$, particularly after being normalized to a probability vector, remains robust, as the eigenproblem's conditioning diverges from the linear system's.

Eigenvector sensitivity, rather than linear system sensitivity, governs $\boldsymbol{x}$'s response to perturbations of $\boldsymbol{P}$. For Markov chains, sensitivity inversely correlates with the spectral gap between the dominant and subdominant eigenvalues of $\boldsymbol{P}$ (Golub and Meyer, 1986; Fetterly et al.; Meyer, 1993; Cho and Meyer, 2000).

Specifically, when α increases, the subdominant eigenvalue λ_2 of $\alpha\boldsymbol{P}$ approaches 1, heightening $\boldsymbol{x}$'s susceptibility to perturbations. Thus, selecting $\alpha = 0.85$ over $\alpha = 0.99$, despite reduced fidelity to the PageRank graph hyperlink structure, enhances computational stability by preserving a sufficient spectral gap (Langville and Meyer, 2004).

A.4 Convergence Rate of PageRank

The PageRank algorithm utilizes the power method to compute the principal eigenvector of the matrix M, with its convergence rate defined as $|\frac{\lambda_2}{\lambda_1}|$ (Golub and Van Loan, 2013). Kamvar and Haveliwala (2003) demonstrates that under mild conditions, $\lambda_2 = \alpha$, which implies that the convergence rate of the power method for M is α. For PageRank, a conventional value of α is 0.85, ensuring that the convergence rate remains rapid even on large-scale graphs.

B Experiments

B.1 Models and Datasets

Tab. 5 presents detailed statistics of the datasets used for the classification task. From each test set, we randomly sampled $5,000$ sentences, except when the total test size was smaller, in which case all samples were retained. To enhance diversity, the sampling strategy ensures a balanced distribution of sentence lengths, maintaining equal representation of those shorter and longer than the test dataset's mode length. It is important to note that all these models and datasets are publicly available on the Hugging Face Hub.

B.2 Implementation Details

Using Algorithm 1 from Sec. 4.2, we first construct the adjacency and transition matrices of the graph generated from the information tensors defined in Sec. 4.1. We then apply the PageRank algorithm using the NetworkX package (Hagberg et al., 2008) with its default parameters to get the importance score of each node. We apply a uniform distribution for the personalization vector due to the absence of prior knowledge or specific information about its characteristics.

For KernelSHAP, Integrated Gradients, and LIME methods, we dedicated 10% of the test fold from each benchmark dataset specifically for hyperparameter tuning using the Optuna package (Akiba et al., 2019). For the other techniques, we sticked to the default hyperparameter settings. Additionally, a comparison of the runtimes among the benchmark methods can be found in Tab. 6.

B.3 Runtime of Proposed Methods

Tab. 6 presents a runtime comparison of various methods for computing feature attributions for each token in the test split of the SST2 dataset. Methods relying solely on raw attention weights, such as RawAtt and Rollout, have the shortest runtimes. In contrast, approaches that involve more complex post-processing steps tend to have longer execution times. As demonstrated in Tab. 6, the runtime of our proposed methods remains on par with other computationally intensive approaches.

Computational Complexity. The time complexity of PR-XAI is $O(nl^2t^2)$, where n denotes the number of PageRank iterations (typically 10–50 for convergence), l is the number of transformer layers, and t is the input sequence length. The l^2t^2 factor arises from constructing the layered transition matrix $\boldsymbol{P}$ of dimension $(t \cdot (l+1)) \times (t \cdot (l+1))$ in Algorithm 1, while k accounts for the power method iterations. For comparison, attention-based methods such as Rollout operate in $O(lt^2)$ (Abnar and Zuidema, 2020), while Attention Flow requires $O(l^2t^4)$ (Abnar and Zuidema, 2020). Perturbation-based methods like Integrated Gradients require m forward-backward passes ($m \approx$ 20–300) (Sundararajan et al., 2017), and KernelShap requires N_s perturbation samples per instance (Lundberg and Lee, 2017). As shown in Tab. 6, the empirical runtime of PR-AG (approximately 4.9 seconds on SST2) remains competitive with gradient-based approaches, while delivering superior faithfulness,

Datasets	# Test Samples	# Classes	Pre-trained Model
SST2	1,821	2	textattack/bert-base-uncased-SST-2
Amazon	5,000	2	fabriceyhc/bert-base-uncased-amazon_polarity
IMDB	5,000	2	fabriceyhc/bert-base-uncased-imdb
Yelp	5,000	2	fabriceyhc/bert-base-uncased-yelp_polarity
AG News	5,000	4	fabriceyhc/bert-base-uncased-ag_news

Table 5: Dataset statistics and pre-trained models used in the experiments. All models use identical BERT-base architecture (109M parameters, 12 layers) ensuring fair comparison across datasets.

Methods	Runtime (seconds)
RawAtt	0.897
Rollout	0.962
Grads	3.443
AttGrads	3.876
CAT	3.651
AttCAT	3.734
PartialLRP	4.457
TransAtt	4.312
LIME	3.054
KernelShap	6.635
IG	4.467
DecompX	2.324
GlobEnc	2.039
PR-A	4.021
PR-G	4.543
PR-AG	4.912

Table 6: Average runtime of methods for the SST2 test split. PR-AG runtime remains competitive with other gradient-based methods, while being slightly slower than simpler attention-based methods, demonstrating reasonable computational cost for superior performance.

particularly on long-form text.

This study has been conducted on a computing device running Ubuntu 20.04.4 LTS. The system is powered by Intel(R) Xeon(R) Platinum 8368 CPUs, which operate at a clock speed of 2.40 GHz. This processor features 12 physical cores and 24 threads, enabling efficient parallel computing and optimized execution of computationally intensive tasks. The graphical computations were handled by an NVIDIA RTX 3090 Ti GPU, equipped with 40 GB of dedicated VRAM, ensuring high-speed processing of deep learning and machine learning workloads. The system is also equipped with 230 GB of dedicated system memory, ensuring smooth and efficient experimentation.

C Additional Results

This appendix presents additional experimental analyses that complement our main findings. We provide detailed performance metrics (App. C.1 and App. C.2), statistical validation through ASO tests (App. C.3), cross-architecture comparisons between BERT, DistilBERT, and RoBERTa models (App. C.4), qualitative layer-wise attribution visualizations (App. C.5), and a focused analysis of the Yelp dataset's unique challenges (App. C.6). All these experiments and analyses further validate PR-XAI's effectiveness as a feature attribution method, while also providing deeper insights into attribution patterns across different model architectures and dataset characteristics. We exclude DecompX and GlobEnc from our experiments in this section as they only support specific variants of encoder-only models.

C.1 Main Metrics

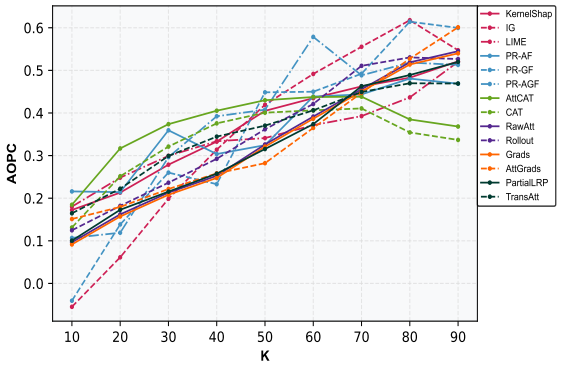
Fig. 2 illustrates a comprehensive comparison of the performance of different feature attribution methods under different corruption rates across three datasets: IMDB, Amazon, and Yelp.

PR-AG consistently outperforms other methods in the IMDB dataset, achieving the highest AOPC scores (peaking at ≈ 0.6) and lowest LOdds scores (reaching below -2.5 at higher k values), indicating a superior ability to identify important tokens in movie reviews.

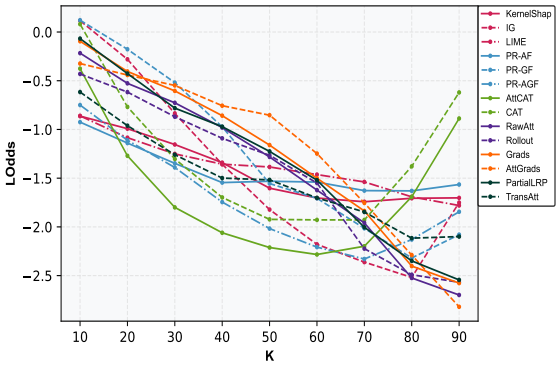
The LOdds scores show a downward trend for most methods as k increases, with PR-AG attaining the best results. TransAtt remains stable, indicating its robustness, while Rollout and RawAtt perform weaker, particularly at higher corruption rates k, suggesting less effective feature attribution.

In the Amazon dataset, PR-AG outperforms other methods, especially at high k values (70%-90%). IG also performs competitively, particularly in LOdds scores. In addition, CAT and AttCAT show significant variability across k values, highlighting their sensitivity to the dataset's semantic and syntactic features.

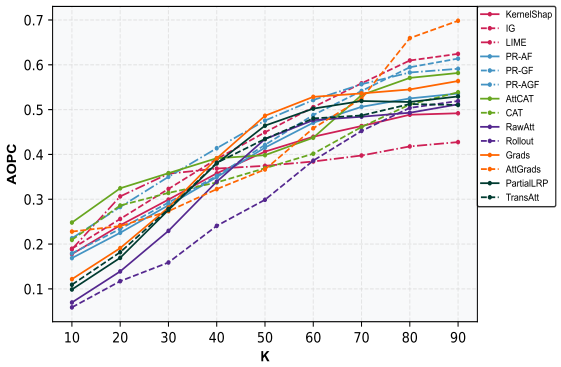
The LOdds scores display a distinctive pattern where most methods reach their lowest values at $k \approx 70\%$, followed by a slight upturn at higher k values. This can indicate an optimal corruption threshold beyond which additional token masking provides diminishing returns. PR-G again shows remarkable consistency, while TransAtt performs moderately well with less variability than others.



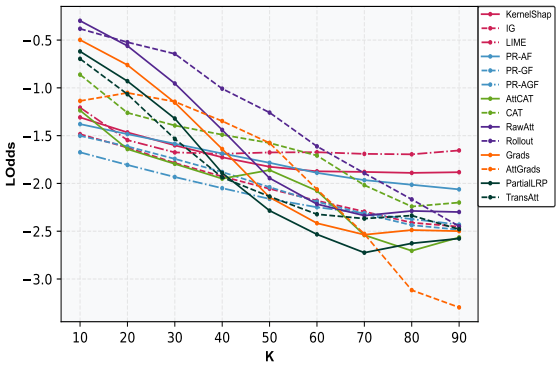
(a) AOPC score for IMDB dataset



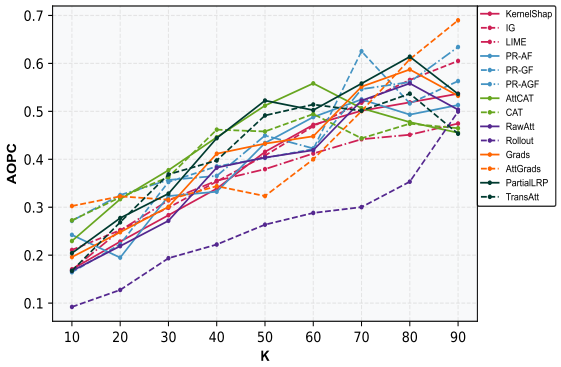
(b) LOdds score for IMDB dataset



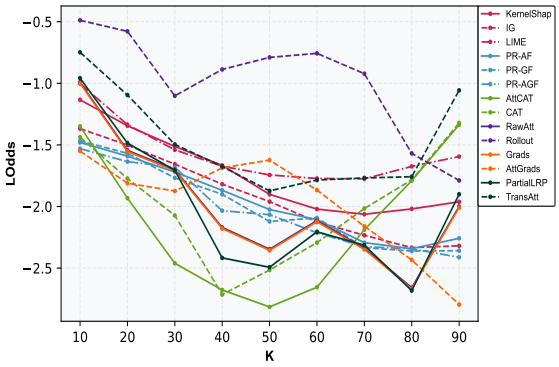
(c) AOPC score for Amazon dataset



(d) LOdds score for Amazon dataset



(e) AOPC score for Yelp dataset



(f) LOdds score for Yelp dataset

Figure 2: AOPC and LOdds scores of different methods in explaining BERT across the varying corruption rates k on IMDB, Amazon, and Yelp datasets. The x-axis illustrates masking the $k\%$ of the tokens in an order of decreasing saliency.

The Yelp dataset reveals unique performance profiles for the evaluated methods. PartialLRP achieves the highest AOPC scores at moderate k values (40%-60%), surpassing even PR-AG in this range, though PR-AG regains dominance at higher k values. PR-AG and AttCAT consistently produce the most favorable LOdds scores among all methods.

C.2 Classification Metrics

The evaluation of feature attribution methods using the classification metrics F1, Accuracy, Precision, and Recall has been summarized in Tab. 7, offering critical insights into how effectively each method identifies tokens that genuinely influence model decisions. PR-AG displays unique performance patterns across datasets with varying characteristics that highlight its strengths and limitations.

Methods	SST2				IMDB				Yelp				Amazon				AG News			
	F1↓	Acc↓	Prec↓	Rec↓	F1↓	Acc↓	Prec↓	Rec↓	F1↓	Acc↓	Prec↓	Rec↓	F1↓	Acc↓	Prec↓	Rec↓	F1↓	Acc↓	Prec↓	Rec↓
RawAtt	0.75	0.75	0.72	0.79	0.69	0.67	0.70	0.69	0.68	0.72	0.74	0.63	0.67	0.68	0.67	0.66	0.65	0.68	0.68	0.68
Rollout	0.81	0.82	0.80	0.82	0.74	0.67	0.65	0.84	0.80	0.83	0.88	0.74	0.71	0.73	0.71	0.72	0.61	0.63	0.64	0.63
Grads	0.78	0.75	0.72	0.79	0.69	0.67	0.70	0.69	0.68	0.72	0.74	0.63	0.67	0.68	0.67	0.66	0.65	0.68	0.68	0.68
AttGrads	0.78	0.78	0.75	0.82	0.76	0.75	0.78	0.76	0.91	0.91	0.89	0.93	0.79	0.82	0.80	0.78	**0.58**	**0.61**	**0.60**	**0.60**
CAT	0.68	0.65	0.61	0.76	0.56	0.49	0.51	0.65	0.70	0.70	0.68	0.73	0.68	0.64	0.67	0.70	0.63	0.64	0.64	0.64
AttCAT	0.68	0.65	0.62	0.76	0.57	0.48	0.50	0.64	0.67	0.66	**0.65**	0.70	0.67	**0.62**	0.66	0.68	0.64	0.64	0.65	0.64
PartialLRP	0.75	0.75	0.71	0.78	0.66	0.65	0.67	0.67	0.65	0.70	0.71	0.61	0.65	0.66	0.65	0.64	0.65	0.68	0.68	0.68
TransAtt	0.73	0.72	0.69	0.76	0.61	0.58	0.60	0.62	**0.62**	0.66	0.66	**0.60**	**0.62**	0.63	0.63	**0.61**	0.63	0.66	0.66	0.66
LIME	0.61	0.63	0.62	0.63	0.55	0.55	0.55	0.55	0.72	0.73	0.72	0.73	0.72	0.73	0.72	0.73	0.67	0.67	0.68	0.67
KernelShap	**0.53**	**0.52**	**0.53**	**0.53**	0.67	0.69	0.68	0.69	0.77	0.78	0.77	0.78	0.67	0.68	0.67	0.68	0.74	0.74	0.75	0.74
IG	0.56	0.57	0.56	0.57	0.48	0.50	0.48	0.50	0.72	0.72	0.72	0.72	0.73	0.74	0.73	0.74	0.67	0.68	0.67	0.67
DecompX	0.66	0.66	0.66	0.67	0.59	0.60	0.58	0.59	0.73	0.74	0.73	0.72	0.69	0.70	0.69	0.69	0.71	0.69	0.70	0.72
GlobEnc	0.70	0.70	0.70	0.69	0.61	0.62	0.60	0.61	0.70	0.71	0.69	0.71	0.71	0.73	0.71	0.71	0.72	0.72	0.73	0.71
PR-A	0.76	0.76	0.77	0.77	0.64	0.66	0.67	0.61	0.77	0.76	0.76	0.77	0.75	0.74	0.74	0.74	0.73	0.72	0.74	0.76
PR-G	0.65	0.66	0.66	0.65	0.55	0.58	0.57	0.58	0.72	0.72	0.71	0.72	0.73	0.77	0.76	0.73	0.69	0.70	0.67	0.70
PR-AG	0.57	0.58	0.56	0.57	**0.42**	**0.48**	**0.41**	**0.48**	0.64	**0.64**	0.66	0.64	0.63	**0.62**	**0.61**	0.63	0.62	0.62	0.61	0.62

Table 7: The average of F1, Accuracy, Precision, and Recall scores of the benchmark methods when we mask **top** $k\%$ tokens. Lower scores are desirable for all metrics (indicated by ↓), indicating a strong ability to mark important tokens. Best results are in bold.

C.2.1 Gradient-Based vs. Attention-Based vs. Hybrid Methods

In examining methods from different categories, PR-AG consistently outperforms its individual components, PR-A and PR-G, across all datasets, and generally, PR-G performs better than PR-A. This clear hierarchy highlights the complementary benefits of combining attention weights with their gradients.

Additionally, token decomposition methods, such as DecompX and GlobEnc, demonstrate better performance than pure attention methods; however, they are still less effective than hybrid approaches like PR-AG and AttCAT. This trend emphasizes that explicitly modeling token interactions through graph-based propagation or attention-gradient aggregation provides greater explanatory power than methods that treat tokens more independently.

C.2.2 Short-form vs. Long-form Text

Although PR-AG demonstrates strong performance on short texts (e.g., SST2), perturbation-based methods such as KernelShap achieve slightly better results. This demonstrates that for shorter texts, where token relationships are more direct, model-agnostic methods can effectively identify important tokens without relying on complex propagation mechanisms.

These findings completely align with the trends observed in Sec. 6.3, indicating that PR-AG's graph-based approach is versatile across various text lengths but particularly excels in capturing long-range dependencies between tokens in longer texts such as those in the IMDB dataset.

C.3 Statistical Significance Test

Utilizing the Almost Stochastic Order (ASO) method (Ulmer et al., 2022; del Barrio et al., 2017), we conducted a statistical significance test by comparing the cumulative distribution functions (CDFs) of two score distributions to evaluate stochastic dominance. Notably, ASO is robust, imposing no assumptions on the score distributions, making it applicable to any metric where higher scores mean better performance.

Dataset	Top $k\%$ Tokens		Bottom $k\%$ Tokens	
	AOPC	LOdds	AOPC	LOdds
SST2	0.065	0.042	0.033	0.042
IMDB	0.051	0.044	0.063	0.051
Yelp	0.631	0.983	0.573	0.917
Amazon	0.057	0.052	0.051	0.057
AG News	0.036	0.042	0.043	0.049

Table 8: ASO test ($\epsilon_{\min}$ values) comparing PR-AG method with the best benchmark for both top (left) and bottom (right) $k\%$ token masking strategies. PR-AG significantly outperforms benchmarks on most datasets, with only Yelp showing non-significant results, proving statistical reliability of main findings.

When comparing model A with model B via the ASO method, we derive the parameter $\epsilon_{\min}$, which serves as an upper bound on the violation of stochastic order. If $\epsilon_{\min} \leq \tau$ (with $\tau \leq 0.5$), model A is deemed stochastically dominant over model B, indicating its superiority. This parameter may also be interpreted as a confidence score, where a lower $\epsilon_{\min}$ reflects greater confidence in the predominance of model A. The null hypothesis for the ASO method is defined as follows:

$$H_0 : \epsilon_{\min} \geq \tau \tag{21}$$

with the significance level α serving as an input parameter that influences ϵ_{min}.

We conducted 1000 independent runs for each method to perform thorough statistical tests, comparing the AOPC and LOdds metrics of our leading proposed method, PR-AG, against top benchmark methods from Tab. 1 and Tab. 2, with $\tau = 0.5$. As detailed in Tab. 8, the PR-AG method consistently outperforms the benchmarks across all datasets, with the exception of Yelp.

C.4 Cross-Architecture Performance Analysis

This section examines the proposed methods on various encoder-only transformer models, such as DistilBERT (Sanh et al., 2020) and RoBERTa (Liu et al., 2019), when applied to the Yelp and SST2 datasets. We compared the output metrics of these models with those of the BERT models discussed in Sec. 6, and documented in Tab. 1 and Tab. 2. Tab. 9 provides the information about the model used for each dataset and each model architecture.

Dataset	Model	Pre-trained Model
SST2	BERT	textattack/bert-base-uncased-SST-2
SST2	DistilBERT	assemblyai/distilbert-base-uncased-sst2
SST2	RoBERTa	syedkhalid076/RoBERTa-Sentimental-Analysis-v1
Yelp	BERT	fabriceyhc/bert-base-uncased-yelp_polarity
Yelp	DistilBERT	neal49/distilbert-yelp
Yelp	RoBERTa	VictorSanh/roberta-base-finetuned-yelp-polarity

Table 9: Pre-trained models used for cross-architecture performance analysis across each dataset.

C.4.1 Top Token Masking Performance

Across the three SST2 model variants in Tab. 10, we observe several noteworthy patterns:

Model Architecture Effects: DistilBERT often performs competitively or even better than its larger counterpart, BERT, despite having fewer parameters. This may indicate that model distillation can improve interpretability by encouraging the model to concentrate on the most important features. On the other hand, RoBERTa typically exhibits lower absolute AOPC values but tends to generate more extreme LOdds scores, meaning that its predictions are highly sensitive to the removal of important tokens.

Consistency Across Models: PR-AG method shows notable consistency, ranking among the top three methods for all model variants, while other methods exhibit more variable performance. This reliability across various architectures is especially valuable for practical applications.

Gradient vs. Attention Methods: Attention-based methods such as RawAtt and Rollout consistently underperform compared to gradient-informed approaches across all model architectures. This supports previous literature suggesting that raw attention weights are insufficient explanations for transformer predictions (Kobayashi et al., 2020; Jain and Wallace, 2019; Serrano and Smith, 2019). It also reveals the significance of backpropagation signals to develop more precise feature attribution methods.

Across the three Yelp model variants in Tab. 10, we observe several noteworthy patterns:

Model Architecture Effect: PartialLRP achieves the highest AOPC on BERT, while AttCAT performs best with DistilBERT, and PR-AG leads with RoBERTa. This variation in AOPC among different model architectures highlights how distinct architectures encode sentiment information through their unique internal mechanisms when applied to the Yelp Dataset.

Consistency Across Models: TransAtt shows the most consistent performance across all three architectures for the Yelp dataset, with relatively stable AOPC and LOdds scores, making it the most reliable method for the Yelp dataset when working with multiple model architectures in production.

Gradient vs. Attention Methods: Methods that incorporate gradient information like AttGrads, AttCAT, and PR-AG consistently outperform their non-gradient counterparts encompassing RawAtt, Rollout, and PR-A across all models, reinforcing the significance of backpropagation signals to develop accurate feature attribution methods.

C.4.2 Bottom Token Masking Performance

Across the three SST2 model variants in Tab. 11, we observe several noteworthy patterns:

Model Architecture Effects: RoBERTa mostly exhibits higher AOPC values when masking less important tokens compared to BERT and DistilBERT models. This suggests that it may rely more evenly on a wider range of input tokens, making the distinction between significant and insignificant tokens less pronounced.

Consistency Across Models: PR-AG displays superior performance across all three SST2 model variants, indicating its ability to accurately identify both important and unimportant tokens, which is a challenging requirement for a reliable feature attribution method.

Methods	BERT		DistilBERT		RoBERTa	
	AOPC↑	LOdds↓	AOPC↑	LOdds↓	AOPC↑	LOdds↓
RawAtt	0.348	-0.973	0.355	-0.995	0.334	-0.951
Rollout	0.322	-0.887	0.307	-0.868	0.318	-0.851
Grads	0.354	-0.313	0.349	-0.324	0.333	-0.338
AttGrads	0.367	-0.654	0.380	-0.685	0.338	-0.631
CAT	0.369	-1.175	0.354	-1.159	0.353	-1.034
AttCAT	0.405	-1.402	**0.411**	**-1.417**	0.378	-1.369
PartialLRP	0.371	-1.171	0.362	-1.185	0.335	-1.017
TransAtt	0.399	-1.286	0.381	-1.264	0.351	-1.226
LIME	0.362	-1.056	0.340	-0.978	0.336	-0.997
KernelShap	0.382	-1.259	0.395	-1.276	0.345	-1.113
IG	0.401	-1.205	0.385	-1.026	0.367	-0.987
PR-A	0.372	-0.991	0.355	-0.784	0.345	-0.771
PR-G	0.365	-0.943	0.351	-0.862	0.369	-0.961
PR-AG	**0.412**	**-1.423**	0.399	-1.406	**0.387**	**-1.384**

(a) SST2 – Top $k\%$ Masking

Methods	BERT		DistilBERT		RoBERTa	
	AOPC↑	LOdds↓	AOPC↑	LOdds↓	AOPC↑	LOdds↓
RawAtt	0.383	-1.985	0.375	-1.956	0.340	-1.886
Rollout	0.260	-0.987	0.252	-0.967	0.267	-1.046
Grads	0.412	-1.994	0.399	-1.819	0.423	-1.824
AttGrads	0.423	-1.978	0.438	-1.973	0.402	-1.896
CAT	0.417	-1.992	0.406	-1.973	0.393	-1.831
AttCAT	0.431	-2.134	**0.446**	**-2.149**	0.403	-1.874
PartialLRP	**0.443**	-2.018	0.431	-1.983	0.391	-1.961
TransAtt	0.411	-1.473	0.395	-1.456	0.377	-1.391
LIME	0.361	-1.568	0.370	-1.588	0.346	-1.494
KernelShap	0.385	-1.736	0.365	-1.714	0.382	-1.692
IG	0.409	-1.924	0.424	-1.942	0.418	**-1.989**
PR-A	0.394	-1.965	0.379	-1.937	0.372	-1.844
PR-G	0.406	-1.997	0.385	-1.721	0.396	-1.964
PR-AG	0.437	**-2.027**	0.434	-1.996	**0.427**	-1.985

(b) Yelp – Top $k\%$ Masking

Table 10: AOPC and LOdds scores of all methods in explaining the transformer-based model across datasets when we mask **top** $k\%$ tokens. Higher AOPC and lower LOdds are desirable, indicating a strong ability to mark important tokens. Best results are in bold.

Methods	BERT		DistilBERT		RoBERTa	
	AOPC↓	LOdds↑	AOPC↓	LOdds↑	AOPC↓	LOdds↑
RawAtt	0.184	-0.693	0.192	-0.707	0.212	-0.733
Rollout	0.221	-0.773	0.235	-0.794	0.236	-0.801
Grads	0.234	-0.776	0.245	-0.798	0.257	-0.815
AttGrads	0.217	-0.713	0.204	-0.731	0.192	-0.705
CAT	0.247	-0.874	0.252	-0.863	0.271	-0.889
AttCAT	0.143	-0.412	0.139	-0.397	0.144	-0.387
PartialLRP	0.163	-0.527	0.180	-0.551	0.192	-0.575
TransAtt	0.148	-0.483	0.154	-0.467	0.133	-0.428
LIME	0.173	-0.603	0.186	-0.616	0.169	-0.576
KernelShap	0.197	-0.729	0.183	-0.682	0.192	-0.713
IG	0.150	-0.532	0.165	-0.552	0.148	-0.538
PR-A	0.191	-0.751	0.206	-0.776	0.217	-0.789
PR-G	0.154	-0.517	0.171	-0.549	0.164	-0.536
PR-AG	**0.122**	**-0.351**	**0.131**	**-0.363**	**0.138**	**-0.384**

(a) SST2 – Bottom $k\%$ Masking

Methods	BERT		DistilBERT		RoBERTa	
	AOPC↓	LOdds↑	AOPC↓	LOdds↑	AOPC↓	LOdds↑
RawAtt	0.157	-0.747	0.169	-0.765	0.182	-0.791
Rollout	0.169	-0.734	0.155	-0.714	0.171	-0.737
Grads	0.131	-0.641	0.140	-0.659	0.143	-0.713
AttGrads	0.127	-0.603	0.139	-0.622	0.146	-0.654
CAT	0.134	-0.659	0.147	-0.674	0.162	-0.686
AttCAT	**0.103**	**-0.339**	0.117	**-0.361**	**0.130**	**-0.373**
PartialLRP	0.116	-0.486	0.126	-0.490	0.158	-0.526
TransAtt	0.123	-0.538	0.134	-0.553	0.128	-0.571
LIME	0.143	-0.687	0.157	-0.719	0.146	-0.704
KernelShap	0.135	-0.645	**0.112**	-0.618	0.152	-0.663
IG	0.130	-0.617	0.144	-0.645	0.143	-0.657
PR-A	0.126	-0.521	0.133	-0.548	0.138	-0.579
PR-G	0.118	-0.417	0.129	-0.449	0.134	-0.457
PR-AG	0.107	-0.394	0.118	-0.421	0.131	-0.449

(b) Yelp– Bottom $k\%$ Masking

Table 11: AOPC and LOdds scores of all methods in explaining the transformer-based model across datasets when we mask **bottom** $k\%$ tokens. Lower AOPC and higher LOdds are desirable, indicating a strong ability to mark important tokens. Best results are in bold.

Gradient vs. Attention Methods: The gap in performance between gradient-based methods and pure attention methods is noticeably smaller with bottom $k\%$ masking than with top $k\%$ masking. This indicates that, although attention weights may struggle to identify the most important tokens, they are still fairly effective at detecting unimportant ones.

Across the three Yelp model variants in Tab. 11, we observe several noteworthy patterns:

Model Architecture Effects: For the Yelp dataset, AttCAT demonstrates the best performance when used with BERT and DistilBERT, while PR-AG performs exceptionally well with RoBERTa. This indicates that AttCAT may be more effective at capturing the specific attention patterns utilized by BERT models in this dataset, whereas PR-AG appears to align better with RoBERTa's internal representation mechanisms.

RoBERTa shows distinct bottom $k\%$ masking behavior compared to BERT variants, with PR-AG achieving significantly better performance than on other architectures. This architectural distinction may stem from RoBERTa's different pretraining approach and optimization, resulting in different token importance distributions.

Consistency Across Models: AttCAT renders remarkable consistency for both BERT and DistilBERT models, consistently holding a top position for these architectures. This reliability indicates that the combination of attention weights and their gradients generates a strong attribution signal that generalizes efficaciously across BERT families.

Gradient vs. Attention Methods: In the Yelp dataset, the difference in performance between pure attention methods, such as RawAtt and Rollout,

and gradient-informed methods like AttCAT and PR-AG is especially noticeable when using bottom $k\%$ masking. The difference in AOPC is consistent across all architectures, indicating that incorporating gradient information significantly enhances the quality of feature attributions, even when it comes to identifying less important tokens.

C.5 Qualitative Visualizations

Fig. 3 demonstrates layer-wise feature attributions computed by the PR-AG method for both BERT and DistilBERT models on the same input sentence: "although this dog is not cute, it is very smart.". This sentence was synthetically constructed by the authors to show contrastive sentiment relationships and received a positive predicted label.

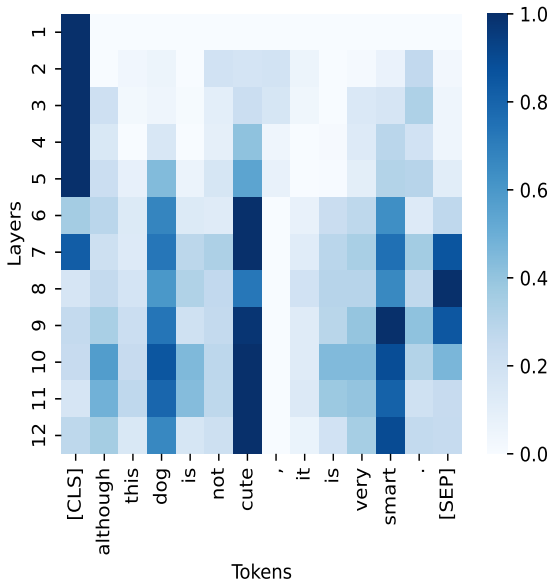
This visualization provides valuable insights into how transformer models of different depths process semantic information and how the PR-AG method effectively captures these attribution patterns across architectures, complementing the prior quantitative findings in Tab. 10 and Tab. 11.

C.5.1 BERT Attribution Analysis

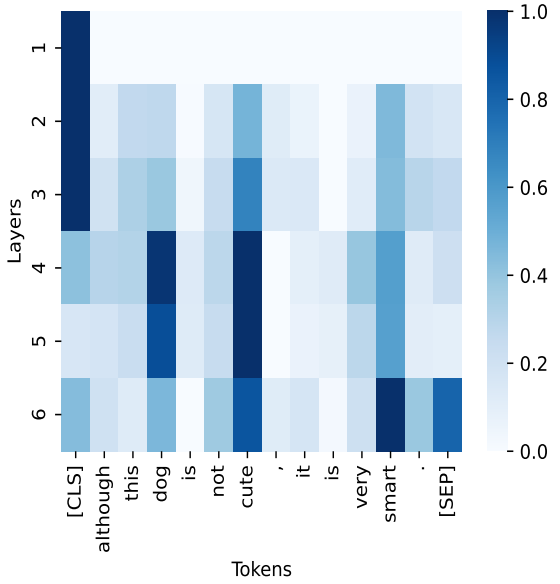
The BERT heatmap exhibits a vivid progression of feature attribution across its 12 layers. Early layers 1-3 predominantly focus on special tokens [CLS] and [SEP] while establishing minimal initial representations of content tokens, suggesting these layers primarily handle structural and positional information. The mid-layers 4-8 display increasing attribution to semantically meaningful tokens, including "dog", "not", "cute", and "smart" indicating progressive development of content understanding. Deep layers 9-12 exhibit the most concentrated and focused attributions for sentiment-determining tokens "not", "cute", and "smart", demonstrating how deeper layers refine semantic representations crucial for classification decisions.

C.5.2 DistilBERT Attribution Analysis

Despite having only 6 layers, compared to BERT's 12 layers, DistilBERT shows remarkably similar attribution patterns, demonstrating highly effective knowledge compression during distillation. This model reaches peak attributions for key semantic tokens by layers 4-5, suggesting faster semantic convergence than BERT—a necessary adaptation to its reduced depth. This model displays more concentrated (darker blue) attributions for tokens "not" and "smart" in its final layers, potentially compensating for fewer layers with more focused feature extraction per layer.



(a) Layer-wise attributions computed by PR-AG for BERT.



(b) Layer-wise attributions computed by PR-AG for DistilBERT.

Figure 3: Layer-wise feature attributions computed by PR-AG for BERT (top) and DistilBERT (bottom). PR-AG captures semantic progression from the structural token [CLS] in early layers to sentiment-critical tokens ("not", "cute", and "smart") in deeper layers, showing architecturally-aware attribution regardless of model depth (BERT vs DistilBERT).

C.5.3 Cross-Model Attribution Analysis

Both models can correctly identify semantically crucial tokens "not", "cute", and "smart", validating that PR-AG effectively captures important tokens

regardless of model architecture. These models assign high importance to [CLS] and [SEP] tokens, particularly in early layers; however, BERT shows more significant reactivation of [CLS] token in later layers, suggesting architectural differences in the information aggregation for text classification.

(a) Feature attributions computed by PR-A for BERT.

(b) Feature attributions computed by PR-G for BERT.

(c) Feature attributions computed by PR-AG for BERT.

Figure 4: Feature attributions computed by PR-A, PR-G, and PR-AG for BERT. PR-AG correctly identifies parallel semantic structures (both "not" instances) with balanced attribution, while PR-A shows almost uniform distribution and PR-G shows inconsistent weighting, proving the hybrid approach's superiority for complex sentence structures.

(a) Feature attributions computed by PR-A for DistilBERT.

(b) Feature attributions computed by PR-G for DistilBERT.

(c) Feature attributions computed by PR-AG for DistilBERT.

Figure 5: Feature attributions computed by PR-A, PR-G, and PR-AG for DistilBERT. PR-AG correctly identifies parallel semantic structures (both "not" instances) with balanced attribution, while PR-A shows almost uniform distribution and PR-G shows inconsistent weighting, proving the hybrid approach's superiority for complex sentence structures.

C.5.4 Comparative Analysis of Proposed Methods

Fig. 4 and Fig. 5 provide a detailed comparison of three proposed attribution methods: PR-A, PR-G, and PR-AG. These methods were applied to the sentence "it is not too fast and not too slow", which was sampled from the SST2 training set having a positive label. This sentence features a balanced negation structure, making it an excellent test case to evaluate attribution methods. Precise identification of parallel semantic elements and the critical role of negation markers are crucial in this context.

PR-A Attribution in BERT (Fig. 4a):
PR-A, which relies solely on attention weights, has several shortcomings when it comes to identifying the significance of individual tokens. In fact, the attribution scores produced by PR-A tend to be uniformly distributed across tokens, which fails to effectively capture the importance of semantically critical words such as "not" in comparison to less significant function words. This lack of distinction doesn't accurately represent the semantic structure of a sentence, especially given the crucial role that negation markers play in shaping the meaning of the sentence.

PR-G Attribution in BERT (Fig. 4b):
PR-G, which leverages gradient information, demonstrates better token discrimination compared to PR-A, but it has notable shortcomings. While it accurately identifies the first instance of "not" as significant, it displays inconsistent attribution between the parallel negation structures, unjustly emphasizing "fast" over "slow." This imbalance indicates that PR-G does not fully recognize the semantic equivalence of the parallel phrases "not too fast" and "not too slow".

PR-AG Attribution in BERT (Fig. 4c):
PR-AG distinctly demonstrates its superiority by effectively identifying the most important tokens. The negation token "not" receives significantly high attribution scores in both cases, indicating BERT's recognition of negation as semantically important. The adjective tokens "fast" and "slow" also receive significant attribution scores, indicating that PR-AG effectively identifies these descriptive terms as key elements. Additionally, the minimal attribution score assigned to functional tokens such as "and", "it" and "is" demonstrates BERT's capability to differentiate between structural and semantically significant components in contrastive expressions.

PR-A Attribution in DistilBERT (Fig. 5a):
When using DistilBERT, the limitations of PR-A become even more apparent. The attributions show an even more uniform pattern compared to BERT, indicating that PR-A has difficulty in identifying important tokens from DistilBERT's compressed representations. This almost uniform distribution

offers little insight into which tokens truly influence the model's comprehension of the sentence.

PR-G Attribution in DistilBERT (Fig. 5b): PR-G improves over PR-A in DistilBERT but still displays unbalanced attribution between parallel structures. It identifies "fast" as highly important while giving less importance to the second negation structure, particularly "slow". This inconsistency shows that PR-G cannot fully capture the semantic symmetry present in the sentence.

PR-AG Attribution in DistilBERT (Fig. 5c): PR-AG maintains its superior performance in the DistilBERT architecture, successfully identifying important tokens. The attribution scores highlight both negation words "not" as highly important, with additional emphasis on the adjectives "fast" and "slow". In contrast, function words such as "and", "it" and "is" receive minimal scores.

The PR-AG method demonstrates remarkable consistency across BERT and DistilBERT, showing its robustness to architectural differences. This suggests that the method can effectively operate across various model sizes and designs.

PR-AG's superior performance stems from its effective combination of attention patterns (PR-A) and gradient information (PR-G). Neither source alone provides complete attribution information; attention weights lack specificity, while gradients alone miss structural balance. By integrating both through the PageRank algorithm, PR-AG creates a more comprehensive token importance distribution.

C.6 Special Case of Yelp Dataset

The analysis of PR-AG's performance on the Yelp dataset reveals an interesting pattern across Tab. 1, Tab. 2, and Tab. 7. Despite demonstrating strong performance in LOdds score when masking top tokens, PR-AG falls short of achieving the highest AOPC score on Yelp, with PartialLRP consistently outperforming it. This finding contrasts with PR-AG's superior performance across all other datasets, where it generally excels in both metrics. This dataset-specific limitation can arise from several interconnected factors:

Contrastive Sentiment Structure: The Yelp reviews typically include mixed and contrastive evaluations, exhibiting both positives and negatives ("fantastic food, but the service leaves a lot to be desired."). This pattern of shifting sentiment often appears multiple times within a single review.

Heavily Qualified Sentiment: The Yelp reviews often include strong modifiers and qualifiers ("absolutely amazing", "somewhat disappointing"), where the modifier significantly impacts sentiment strength.

Domain-Specific Sentiment Terms: The Yelp reviews include industry-specific sentiment terms ("overpriced", "underwhelming", "authentic") in contrast to general sentiment words found in other datasets.

In addition, examining the results of masking bottom tokens in Tab. 2, we observe that AttCAT consistently outperforms PR-AG on both metrics for the Yelp dataset. This suggests that AttCAT's specific attention-gradient integration may better identify the importance hierarchy in sentiment-rich text. Based on Tab. 11, this pattern holds across model architectures like BERT, DistilBERT, and RoBERTa, indicating a significant characteristic of the dataset rather than a model-specific limitation.

Furthermore, the classification metrics in Tab. 7 demonstrate that gradient-informed methods, such as AttGrads, achieve remarkably high F1 scores and accuracy on the Yelp dataset, suggesting that the distribution of feature importance for this dataset follows patterns where direct gradient information provides stronger signals than PR-AG's PageRank-based diffusion of importance.

These findings suggest that, although PR-AG provides a strong general-purpose framework for feature attribution, the unique characteristics of each dataset can still affect the best selection of attribution method.