

# XAI for Transformers: Better Explanations through Conservative Propagation

Ameen Ali<sup>1</sup>, Thomas Schnake<sup>2,3</sup>, Oliver Eberle<sup>2,3</sup>, Grégoire Montavon<sup>2,3</sup>  
 Klaus-Robert Müller<sup>2,3,4,5</sup>, Lior Wolf<sup>1</sup>

<sup>1</sup> The School of Computer Science, Tel-Aviv University, Israel

<sup>2</sup> Machine Learning Group, Technische Universität Berlin, Berlin, Germany

<sup>3</sup> Berlin Institute for the Foundations of Learning and Data – BIFOLD, Berlin, Germany

<sup>4</sup> Department of Artificial Intelligence, Korea University, Seoul, Korea

<sup>5</sup> Max Planck Institute for Informatics, Saarbrücken, Germany

## Abstract

Transformers have become an important workhorse of machine learning, with numerous applications. This necessitates the development of reliable methods for increasing their transparency. Multiple interpretability methods, often based on gradient information, have been proposed. We show that the gradient in a Transformer reflects the function only locally, and thus fails to reliably identify the contribution of input features to the prediction. We identify Attention Heads and LayerNorm as main reasons for such unreliable explanations and propose a more stable way for propagation through these layers. Our proposal, which can be seen as a proper extension of the well-established LRP method to Transformers, is shown both theoretically and empirically to overcome the deficiency of a simple gradient-based approach, and achieves state-of-the-art explanation performance on a broad range of Transformer models and datasets.

## 1 Introduction

Transformer models Vaswani et al. [2017] have attracted increasing interest and shown excellent performance in domains such as natural language processing (NLP) Vaswani et al. [2017], Devlin et al. [2019], Radford et al. [2019], vision Dosovitskiy et al. [2021] or graphs Ying et al. [2021], Yun et al. [2019]. Yet, their typically very high complexity (up to billions of parameters Narayanan et al. [2021]) makes these models notoriously intransparent and their predictions inaccessible to the user. Since Transformer models have heavy application in potentially sensitive domains, e.g. as support in recruiting processes Aleisa et al. [2021], the development of methods that explain their decisions is essential, since they allow one to verify whether the model makes fair decisions and does not discriminate protected classes Bolukbasi et al. [2016], Gonen and Goldberg [2019].

In this work, we ask how to bring Explainable AI to Transformers in a theoretically sound manner, focusing in particular on the axiom of conservation, which underlies a number of popular explanation techniques Bach et al. [2015], Lundberg and Lee [2017], Sundararajan et al. [2017]. We embed our analysis in the Layer-wise Relevance Propagation (LRP) framework Bach et al. [2015], which allow us to analyze conservation at the level of individual modules and layers of the Transformer model. Our analysis reveals that the conservation properties of existing XAI techniques can be severely impaired when extended to Transformer models. We propose a theoretically grounded, and easy to implement countermeasure, consisting of strategically ‘detaching’ part of the Transformer model’s forward computation before computing explanations.

We provide qualitative and quantitative experiments on different data domains, such as natural language understanding, computer vision and graph analysis. Comparison is performed with state-of-the-art baseline explanation methods for Transformers. For the quantitative analysis, we perform different input perturbation schemes, in which we track the behavior of the model when relevant or irrelevant features are added to or removed from the input data sample. The proposed explanation method for Transformers presents excellent performance on the qualitative and quantitative experiments and outperforms all existing baseline methods in most of the tasks. Hence, our fairly intuitive explanation method provides both theoretical soundness in its formal derivation and leading performance on many important data domains. Our code is publicly available <sup>1</sup>

## 2 Related Work

The approaches for Transformer explanation can be divided into (1) methods that extract the attention heads from

<sup>1</sup>[https://github.com/AmeenAli/XAI\\_Transformers](https://github.com/AmeenAli/XAI_Transformers)

each Transformer block, using them for interpretation, (2) gradient-based methods, and (3) perturbation-based methods. We would like to mention a few of these previous works.

The extraction of the attention heads from an attention module, to understand or visualize the prediction strategy of the model, has a long history Bahdanau et al. [2015], also for non-Transformer architectures, yet it is known to be myopic and unreliable Jain and Wallace [2019]. Besides the simple extraction of the raw attention weights, there have been attempts to use the attention heads for defining more elaborate explanation mechanisms, such as ‘Attention Rollout’ and ‘Attention Flow’ Abnar and Zuidema [2020].

Other authors incorporate gradient methods to explain Transformer models, such as integrated gradient Wallace et al. [2019] or input gradients Atanasova et al. [2020]. The gradient methods Saliency, Gradient  $\times$  Input (GI) or Guided Backpropagation has already been applied in numerous models and domains, found their use in Transformers, as well Atanasova et al. [2020].

In particular, there have been multiple attempts to implement the layer-wise relevance propagation (LRP) method Bach et al. [2015] in Transformers Voita et al. [2019] and other attention-based models Ding et al. [2017]. In addition, LRP has been applied to explain predictions of other models on NLP tasks Wu and Ong [2021], such as BERT Devlin et al. [2019]. Other approaches to gradient propagation in Transformer blocks can be found in Chefer et al. [2021a,b], where the relevancy scores are obtained by combining attention scores with LRP or attention gradients.

There are also a few examples in which perturbation-based methods used input reductions Feng et al. [2018], Prabhakaran et al. [2019] to determine the most relevant parts of the input, by observing change in model confidence or Shapley values Lundberg and Lee [2017], Atanasova et al. [2020].

The experiments we conduct employ both the conventional Transformer, as well as a Transformers on graphs. Specifically, we employ the recent Graphomer model of Ying et al. [2021]. Other graph Transformers include Yun et al. [2019], Yoo et al. [2020], Maziarka et al. [2020], Zhao et al. [2021]. A simple explainability procedure based on the obtained attention maps is presented by Yun et al. [2019].

Previous works on explainability in graph neural networks include gradient-based methods Pope et al. [2019], graph decomposition Schnake et al. [2021] graph perturbations Ying et al. [2019], Luo et al. [2020], and a local approximation using simpler models Huang et al. [2020].

For a more general review of explainable AI, we refer to the survey papers Montavon et al. [2018], Guidotti et al. [2019], Samek et al. [2019], Danilevsky et al. [2020], Arrieta et al. [2020], Samek et al. [2021].

### 3 A Theoretical View on Explaining Transformers

Very recently, techniques such as Gradient  $\times$  Input (GI) Shrikumar et al. [2017], Ancona et al. [2019] or Layer-wise Relevance Propagation (LRP) Bach et al. [2015] have been extended to explain Transformers Voita et al. [2019], Wu and Ong [2021]. New explanation techniques, such as attention rollouts Abnar and Zuidema [2020], or other generic ways to aggregate attention information Chefer et al. [2021a,b] have also been developed for a similar purpose. These recent works have provided empirical evidence that Transformers can be made explainable to a significant extent.

In this section, we aim to gain a better theoretical understanding of the problem of explaining Transformers, drawing on the same ‘axiomatic approach’ Shapley [1953], Strumbelj and Kononenko [2010], Sundararajan et al. [2017], Montavon [2019], Hesse et al. [2021] used for analyzing and developing Explainable AI in the context of standard deep neural networks. A main axiom employed in Explainable AI, especially for the task of attribution, is *conservation*. The conservation axiom (also known as *completeness* or *efficiency*) states that scores assigned to input variables and forming the explanation must sum to the output of the network, in other words, each input variable contributes a share of the predicted score at the output. Workhorses of XAI, such as Layer-wise Relevance Propagation Bach et al. [2015], Gradient  $\times$  Input Shrikumar et al. [2017], Srinivas and Fleuret [2019], Hesse et al. [2021]), Integrated Gradients Sundararajan et al. [2017] or Shapley Values Strumbelj and Kononenko [2010], Lundberg and Lee [2017], are either designed to satisfy this conservation axiom, or can be derived from it directly.

The Layer-wise Relevance Propagation framework Bach et al. [2015] considers a particularly strong form of conservation, where each layer, component, or even neuron in the network is subject to the conservation axiom. In particular, the ‘relevance’ received by a given layer/component/neuron from the layer above must be fully redistributed to the layer below. Many relevance propagation rules have been developed within the LRP framework to address the specificities of different data and architectures. Interestingly, the rudimentary but popular Gradient  $\times$  Input method, although commonly viewed as a gradient-based method, can also be embedded in the LRP framework Shrikumar et al. [2017], Ancona et al. [2019], Montavon [2019], and potentially be used as a starting point (or default choice) for setting propagation rules. Furthermore, with this embedding into LRP, one can perform a detailed analysis of the GI explanation procedure, identifying layers or components where the conservation breaks, and possibly derive better propagation rules for these layers in a second step.

Denote by  $(x_i)_i$  and  $(y_j)_j$  the vectors of neurons representing the input and output of some layer or component of interest in a neural network, and by  $f$  the output of the

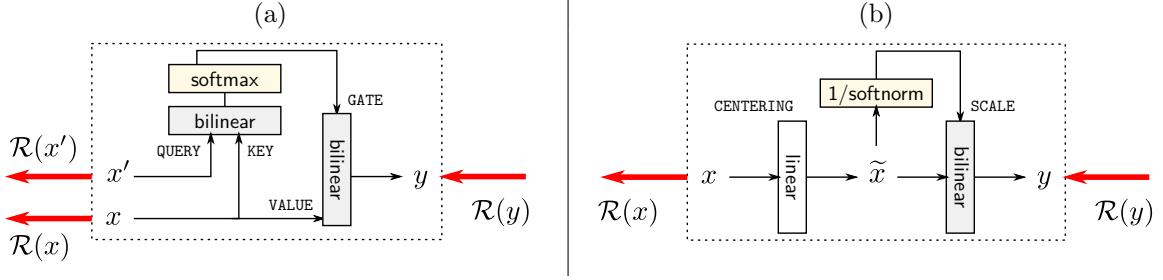


Figure 1: Diagram of specific modules found in Transformer architectures: (a) attention head, (b) LayerNorm. Modules are depicted with the types of nonlinearities they incorporate (bilinear form, softmax, division by the norm), and annotated with the variables used in this paper. Red arrows denote the relevance propagation process under study.

neural network. Gradient  $\times$  Input attributions on these two vector representations can be computed as:

$$\mathcal{R}(x_i) = x_i \cdot (\partial f / \partial x_i) \quad (1)$$

$$\mathcal{R}(y_j) = y_j \cdot (\partial f / \partial y_j) \quad (2)$$

respectively. Recall that the gradients at different layers of a neural network are related via the chain rule as follows:

$$\frac{\partial f}{\partial x_i} = \sum_j \frac{\partial y_j}{\partial x_i} \frac{\partial f}{\partial y_j} \quad (3)$$

Injecting (1) and (2) into (3), the gradient propagation rule can be converted into an equivalent relevance propagation rule:

$$\mathcal{R}(x_i) = \sum_j \frac{\partial y_j}{\partial x_i} \frac{x_i}{y_j} \mathcal{R}(y_j) \quad (4)$$

where we use the convention  $0/0 = 0$ . This is the LRP view on GI. With such interpretation, we can study whether GI is conservative layer-wise or at the level of the given component, by testing whether  $\sum_i \mathcal{R}(x_i) = \sum_j \mathcal{R}(y_j)$ . If this holds for every component of the neural network, then conservation also holds globally. We will now show that two components of the Transformer cause a significant break of conservation and therefore require an improved propagation rule.

### 3.1 Propagation in Attention Heads

Let us consider the attention head, which makes use of a Query-Key-Value mechanism, and which is a core component of Transformers. As defined in Vaswani et al. [2017], attention heads have the structure

$$Y = \text{softmax}\left(\frac{1}{\sqrt{d_K}}(X'W_Q)(XW_K)^\top\right)X \quad (5)$$

where  $X = (x_i)_i$  and  $X' = (x'_j)_j$  are the input sequences of token embeddings,  $Y = (y_j)_j$  is the sequence of output embeddings,  $d_K$  is the dimensionality of the Key-vector, and  $W_{\{Q,K,V\}}$  are learned projection matrices. In the equation above, we omitted the multiplication by the embedding  $W_V$ , since the latter can be viewed as a subsequent linear layer

and can therefore be treated separately. For the purposes of our analysis, we rewrite Eq. (5) as:

$$y_j = \sum_i x_i p_{ij} \quad (6)$$

where

$$p_{ij} = \frac{\exp(q_{ij})}{\sum_{i'} \exp(q_{i'j})}$$

is the softmax computation and  $q_{ij} = \frac{1}{\sqrt{d_K}}x_i^\top W_K W_Q^\top x'_j$  is the matching function between tokens of the two input sequences. Note that the output of Eq. (6) depends on the input tokens both explicitly, via the term  $x_i$ , and via the gating term  $p_{ij}$ , which itself depends on  $x$  and  $x'$ . Also, we observe that for each index  $j$ , we have an associated distribution  $p_j = (p_{ij})_i$ , and we denote by  $\mathbb{E}_j[\cdot]$  and  $\text{Cov}_j(\cdot, \cdot)$ , respectively, the expectation and covariance over this distribution, e.g.  $\mathbb{E}_j[x] = \sum_i x_i p_{ij}$ .

We analyze the relevance propagation associated with applying Gradient  $\times$  Input to the Transformer model. For this, we define input token relevance as  $\mathcal{R}(x_i) = x_i^\top (\partial f / \partial x_i)$  and  $\mathcal{R}(x'_j) = x'_j^\top (\partial f / \partial x'_j)$ , and output token relevance as  $\mathcal{R}(y_j) = y_j^\top (\partial f / \partial y_j)$ .

**Proposition 1.** *Assume that the inputs to the attention head and its softmax gate are centered, i.e.  $\mathbb{E}_j[q_{:j}] = 0$  and  $\mathbb{E}_j[x] = 0$  under all probability distributions  $p_j$ . The relevance propagation associated with Gradient  $\times$  Input is characterized by the conservation equation:*

$$\begin{aligned} \sum_i \mathcal{R}(x_i) + \sum_j \mathcal{R}(x'_j) \\ = \sum_j \mathcal{R}(y_j) + 2 \sum_j \text{Cov}_j(q_{:j}, x)^\top \frac{\partial f}{\partial y_j} \end{aligned} \quad (7)$$

*Which implies that conservation breaks if the rightmost term is non-zero.*

The proof can be found in Appendix A. Prop 1 implies that conservation between layers may not hold in the presence of covariates between  $q_{:j}$  and  $x$ , something that is likely to occur, since the former is a function of the latter. Note that the centering assumption we made in the Proposition

above mainly serves to arrive at a simple and intuitive conservation equation. We provide the more general equation in Appendix A.

To address the lack of conservation in Gradient  $\times$  Input (and the resulting consequence that some attention heads may be over- or underrepresented in the explanation), we will propose in Section 4 an alternative propagation rule that retains the conservation property and also works better empirically.

### 3.2 Propagation in LayerNorm

Another layer that is rather specific to the Transformer is ‘LayerNorm’. For our analysis, we focus on the core part of LayerNorm, consisting of centering and standardization (and ignore the subsequent affine transformation):

$$y_i = \frac{x_i - \mathbb{E}[x]}{\sqrt{\epsilon + \text{Var}[x]}} \quad (8)$$

Here,  $\mathbb{E}[\cdot]$  and  $\text{Var}[\cdot]$  denote the mean and variance over all activations of the corresponding channel (and potentially minibatch).

**Proposition 2.** *The relevance propagation associated with Gradient  $\times$  Input is characterized by the conservation equation:*

$$\sum_i \mathcal{R}(x_i) = \left(1 - \frac{\text{Var}[x]}{\epsilon + \text{Var}[x]}\right) \sum_i \mathcal{R}(y_i) \quad (9)$$

The proof is given in Appendix B. We observe that conservation is never satisfied. Conservation breaks especially strongly when  $\epsilon$  is small compared to  $\text{Var}[x]$ . This relevance collapse can be observed in Fig. 2.

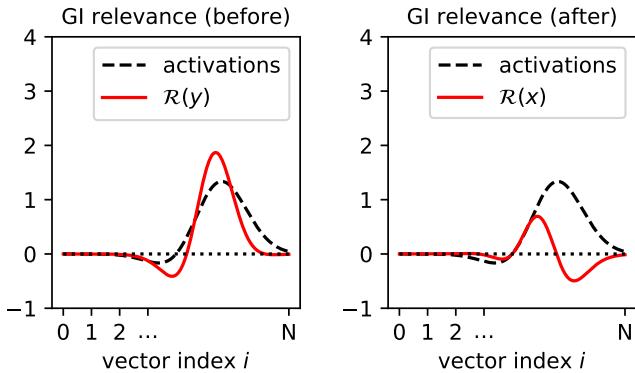


Figure 2: Comparison of the relevance associated with Gradient  $\times$  Input before and after propagation in LayerNorm. We can observe a ‘relevance collapse’ (visible on the right of the peak activation).

## 4 Better LRP Rules for Transformers

We propose to address the deficiencies of Gradient  $\times$  Input revealed by our layer-wise analysis. Specifically, we take the LRP view on Gradient  $\times$  Input as a starting point (Eq. (4)) and replace the implicit propagation rules in attention heads and LayerNorm (two components we identified as breaking conservation) by ad-hoc propagation rules that are conservative by design.

Specifically, we propose at explanation time to make a locally linear approximation of the *attention head* by viewing the gating terms  $p_{ij}$  as constants. Consequently, these terms can be interpreted as the weights of a linear layer locally mapping the input sequence  $x$  to the output sequence  $y$ . We can then use the canonical LRP rule for linear layers defined as

$$\mathcal{R}(x_i) = \sum_j \frac{x_i p_{ij}}{\sum_{i'} x_{i'} p_{i'j}} \mathcal{R}(y_j) \quad (\text{AH-rule})$$

to propagate the relevance scores from the layer output to the layer input. With such a reformulation, we also note that the query sequence  $x'$  appears disconnected, and consequently, we have implicitly  $\mathcal{R}(x'_j) = 0$ . This strategy has also been used and justified theoretically for LSTM blocks Arras et al. [2019], and has been shown empirically to yield superior performance to gradient-based methods, in particular Gradient  $\times$  Input.

Furthermore, to address the particularly severe break of conservation in *LayerNorm*, we propose again a locally linear approximation at explanation time, by viewing the multiplicative factor  $\alpha = (\sqrt{\epsilon + \text{Var}[x]})^{-1}$  as constant. The entire LayerNorm operation can then be expressed by the linear transformation  $\alpha C x$  where  $C = I_N - \frac{1}{N} \mathbf{1}_{(N \times N)}$  is the ‘centering matrix’ and where  $N$  is the number of tokens in the input sequence. Using again the same canonical LRP rule for linear layers, this time with weights  $\alpha C$ , we obtain:

$$\mathcal{R}(x_i) = \sum_j \frac{x_i \cdot (\delta_{ij} - \frac{1}{N})}{\sum_{i'} x_{i'} \cdot (\delta_{i'j} - \frac{1}{N})} \mathcal{R}(y_j) \quad (\text{LN-rule})$$

where  $(\delta_{ij} - \frac{1}{N})$  is another way of writing  $C_{ij}$  and the factor  $\alpha$  present in the numerator and denominator cancels out.

In practice, these rules do not need to be implemented explicitly. We can simply observe that they are effectively the same rules as those induced by Gradient  $\times$  Input, with the gating and rescaling terms in their respective layers treated as constant (i.e. detached in the forward computation so that the gradient does not propagate through it). We therefore propose the following implementation trick for the rules above:

**Implementation Trick:** To compute the proposed improved LRP explanation, rewrite Eq. (6) as

$$y_j = \sum_i x_i [p_{ij}] . \text{detach}()$$

in every attention head, and rewrite Eq. (8) as

$$y_i = \frac{x_i - \mathbb{E}[x]}{\left[ \sqrt{\epsilon + \text{Var}[x]} \right] . \text{detach}()}$$

in every LayerNorm. Then extract the LRP explanation by calling Gradient  $\times$  Input on the resulting function  $f$ .

This trick makes the method straight-forward to implement, as it simply consists of adding `detach()` calls at the appropriate locations in the neural network code, and then running standard Gradient  $\times$  Input. Furthermore, the runtime is at least as good as Gradient  $\times$  Input, and typically better, due to the resulting simplification of the gradient computation.

## 5 Experimental Setup

The proposed approach is tested on several Transformer models trained on various datasets. We benchmark the performance of our method against a number of other approaches proposed in the literature and commonly used for explaining Transformer-type architectures.

### 5.1 Datasets

We use the following datasets from natural language processing, image classification, as well as molecular modeling to evaluate the different XAI approaches. For the NLP experiments, we consider sentiment classification on the SST-2 Socher et al. [2013] and IMDB datasets Maas et al. [2011] which contain 11,844 and 50,000 movie reviews, respectively, for binary classification into negative or positive sentiment. In addition, we use the TweetEval Dataset Xiong et al. [2019] for tweet classification on sentiment (59,899), hate detection (12,970) and emotion recognition (5,052). Furthermore, the SILICONE Dataset Chapuis et al. [2020] is also used for the tasks of emotion detection (Semaine 13,708) and utterance sentiment analysis (Meld-S 5,627).

For experiments with graph Transformers, the MNIST superpixels data Monti et al. [2017] is extracted from 70,000 samples for digit classification. Each node represents one image patch (superpixel), and is connected to its neighbors. Another graph dataset is based on the MoleculeNet Wu et al. [2018] benchmark: the BACE dataset contains 1,522 compounds, together with their structures and binary labels indicating a binding result for a set of inhibitors of human  $\beta$ -secretase 1 (BACE-1) Subramanian et al. [2016].

## 5.2 Benchmark Methods

First, we compare to a ‘Gradient  $\times$  Input’ Denil et al. [2014], Shrikumar et al. [2017], Atanasova et al. [2020] baseline without considering any modifications to the individual gradient computations as described in Section 3.

In addition, we compute averages over last-layer attention head vectors (‘Attention-last’) as proposed in Hollenstein and Beinborn [2021] as well as ‘Rollout’ and attention flow (‘A-flow’) Abnar and Zuidema [2020] which capture the layer-wise structure of deep Transformer models in comparison to raw attention head analysis.

‘Generic Attention Explainability’ (GAE) by Chefer et al. [2021a] propagates attention gradients together with gradients from other parts of the network, resulting in state-of-the art performance in explaining Transformer architectures.

We consider two variants of our proposed LRP-based technique. First ‘LRP (AH)’ where propagation through attention heads is handled via the AH-rule described in Section 4. For any other layers, we use the GI-equivalent propagation rule (i.e. in practice simply propagating gradient without detaching terms). The second variant, which we call ‘LRP (AH+LN)’ additionally propagates through the LayerNorm using the LN-Rule.

## 6 Results

We now turn to evaluating the LRP-based method we have proposed. We first validate that conservation is indeed maintained, and then continue with quantitative perturbation experiments. Finally, we visualize the different explanations produced by these methods qualitatively.

### 6.1 Conservation

As a starting point, we would like to verify whether the insights produced in Section 3 on the conservation, or lack thereof, also hold empirically. We consider for this our two Transformer models trained on the SST-2 and MNIST datasets, and compute GI and LRP (AH+LN) explanations. We then compare the score produced at the output of the Transformer network against the sum of explanation scores over input features of the network. For this purpose, the input features are the positionally encoded embedding vectors present in the first layer.

The results are shown in Fig. 3. Evidently, our LRP approach produces explanations that reflect the output score much more closely than GI, although mild breaks of conservation still occur (likely due to the presence of non-attributable biases in linear layers). Surprisingly, GI explanation scores on MNIST are almost anticorrelated with the output of the model. This clearly highlights a major problem with interpreting GI explanations as attributions of the function output in the context of Transformers.

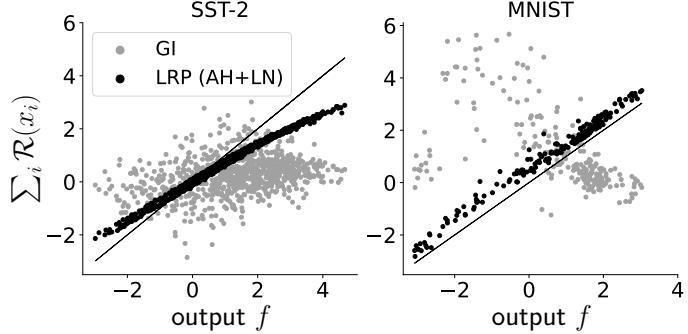


Figure 3: Conservation, or lack thereof, of GI and LRP attributions on two Transformer models. The two axes represent the output score and the sum of explanation scores over the neural network input variables. Each point in the scatter plot represents one example. The closer the points to the diagonal, the more conservative the explanation technique.

## 6.2 Quantitative Evaluation

We now test the performance of different explanation methods using an input perturbation scheme in which the most or least relevant input nodes are considered Schnake et al. [2021]. Two different settings are considered for the Graph Transformer experiments. For the activation task, a good explanation gives an ordering from most to least relevant nodes that when added to an empty graph activate the network output maximally and as quickly as possible. Thus, we observe the output probability  $p_c(x)$  of the correct class  $c$  and report the area under the activation curve (AUAC) with higher AUAC indicating a more faithful explanation with regards to the correct prediction.

In the pruning task, we start with the original graph and remove nodes in the order from smallest to largest absolute values. We measure AU-MSE, which is the area under the mean squared error  $(y_0 - y_{m_t})^2$ , with the model output logits of the unpruned model  $y_0$  and  $y_{m_t}$  representing the output graph after applying the masking  $m_t$  at step  $t$  to the input graph. A lower AU-MSE is desired and indicates that removing less relevant nodes has little effect on the model prediction.

For the NLP experiments, we consider token sequences instead of graph nodes. The activation task starts with an empty sentence of “UNK” tokens, which are then gradually replaced with the original tokens in the order of highest to lowest relevancy. In the pruning task, we remove tokens from lowest to highest absolute relevance by replacing them with “UNK” tokens, similarly to the ablation experiments of Abnar and Zuidema [2020].

In Table 1 and 2, we report results for the activation and pruning tasks and observe that the proposed handling of the gradient in the attention head (AH) and in layer normalization (LN) during backpropagation indeed results in consistently better performance across all datasets. We see

the greatest improvement when applying both the proposed detaching of gating and rescaling terms (AH+LN) together. In addition, explanations based on gradient information are superior to raw attention-based methods (A-Last, Rollout, A-Flow). In Figure 4 we show activation and pruning curves for the SST-2 dataset in a Transformer and the BACE dataset on a Graphomer model and include them for all other datasets in Appendix F. We note that the application of the proposed gradient rules leads to a gradual improvement over naive gradient implementations of GI, especially during the transition from very relevant to rather relevant inputs for the activation task. This suggests that the improved LRP explanations are both sparser and more effective at determining the most relevant input nodes, while attributing low relevance to task-irrelevant input nodes. We will inspect this effect qualitatively in the next section.

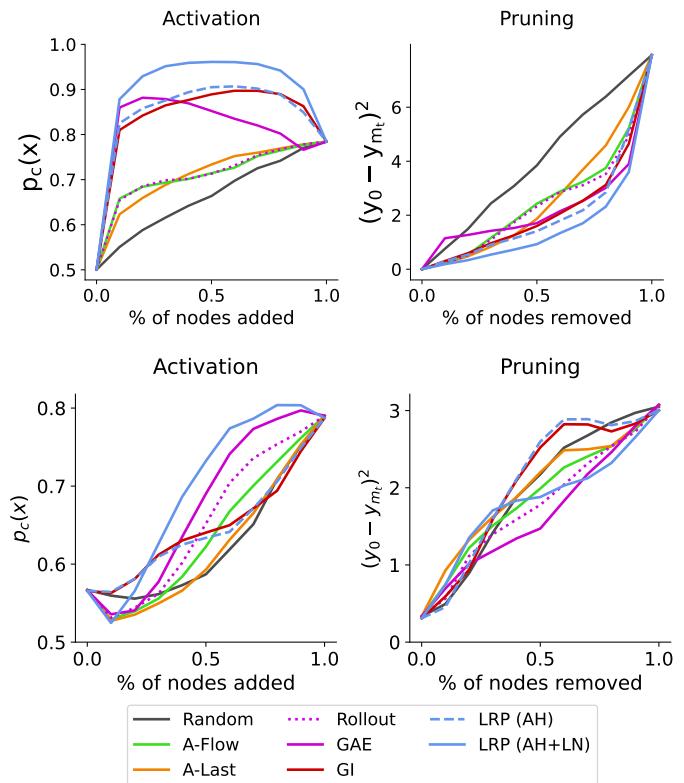


Figure 4: Evaluation of explanations using input perturbations. Results for SST-2 (top row) and BACE (lower row) are shown for the activation task, in which the most relevant nodes are added first (left column), and for pruning, in which the nodes of least absolute relevance are removed first (right column).

## 6.3 Qualitative Results

In this section we will examine the explanations produced with different methods qualitatively for both Transformer and Graphomers.

Table 1: AUAC - area under activation curve. Higher AUAC is better and indicates that adding the most relevant nodes strongly activates the correct model prediction. A-Flow is a computationally very expensive method and was therefore omitted for some methods.

Method	IMDB	SST-2	BACE	MNIST	T-Emotions	T-Hate	T-Sentiment	Meld-S	Semaine
Random	.673	.664	.624	.324	.516	.640	.484	.460	.432
A-Last	.708	.712	.620	.862	.542	.663	.515	.483	.451
A-Flow	-	.711	.637	-	-	-	-	-	-
Rollout	.738	.713	.653	.358	.554	.659	.520	.489	.441
GAE	.872	.821	.675	.426	.675	.762	.611	.548	.532
GI	.920	.847	.646	.942	.652	.772	.651	.591	.529
LRP(AH)	.911	.855	.645	.942	.675	.797	.668	.594	.544
LRP(AH+LN)	<b>.939</b>	<b>.908</b>	<b>.707</b>	<b>.948</b>	<b>.750</b>	<b>.838</b>	<b>.713</b>	<b>.635</b>	<b>.606</b>

Table 2: AU-MSE - area under the mean squared error ( $y_0 - y_{m_t}$ )<sup>2</sup> curve for pruning. Lower is better and indicates that removing less relevant nodes has little effect on the model prediction.

Method	IMDB	SST-2	BACE	MNIST	T-Emotions	T-Hate	T-Sentiment	Meld-S	Semaine
Random	2.16	3.97	1.95	69.82	4.25	9.12	2.87	2.54	1.92
A-Last	1.65	2.56	1.99	45.82	3.73	7.77	1.90	1.74	1.42
A-Flow	-	2.52	1.87	-	-	-	-	-	-
Rollout	1.04	2.43	1.77	115.2	2.85	6.55	1.71	1.53	1.40
GAE	1.63	2.26	1.66	59.81	2.21	7.40	1.61	1.56	1.37
GI	0.87	2.10	2.06	18.06	2.09	6.69	1.41	1.57	1.43
LRP(AH)	0.77	2.02	2.08	18.03	1.83	6.43	1.43	1.69	1.38
LRP(AH+LN)	<b>0.65</b>	<b>1.56</b>	<b>1.61</b>	<b>17.49</b>	<b>1.47</b>	<b>4.88</b>	<b>1.23</b>	<b>1.48</b>	<b>1.08</b>

In Figure 5 we see the relevance scores of different attribution methods when interpreting the prediction of the Transformer model on a sentence from the SST-2 dataset with positive sentiment. We provide additional examples in Appendix D. We can see that most of the methods show that the model focuses on the words “best” and “virtues”, which is a reasonable strategy, since they contribute clearly to a positive sentiment. It is also visible that ‘A-last’ has a more distinct positive or negative focus on the token “eastwood” which suggests an over-confidence in names or entities (we will return to this in section 6.4). Our proposed methods, ‘LRP (AH)’ and ‘LRP (AH+LN)’, consider this entity token to be less relevant for the task (mild coloring) and instead highlight tokens that provide a more general strategy for detecting positive sentiment in a more pronounced manner.

The explanations obtained for a Graphomer model trained on MNIST superpixels are depicted in Figure 6 and we pro-

A-last	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
A-Flow	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
Rollout	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
GAE	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
GI	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
LRP (AH)	[CLS] has a lot of the virtues of eastwood at his best. [SEP]
LRP (AH+LN)	[CLS] has a lot of the virtues of eastwood at his best. [SEP]

Figure 5: Visualisation of the attribution scores per word in a sentence of the SST-2 dataset when predicting the sentiment of the sentence. Opaque red color indicates that the word contributes positively to the prediction, blue indicates that the word has a negative contribution. The heatmaps of the methods ‘A-last’, ‘A-flow’ and ‘Rollout’ are limited to non-negative scores.

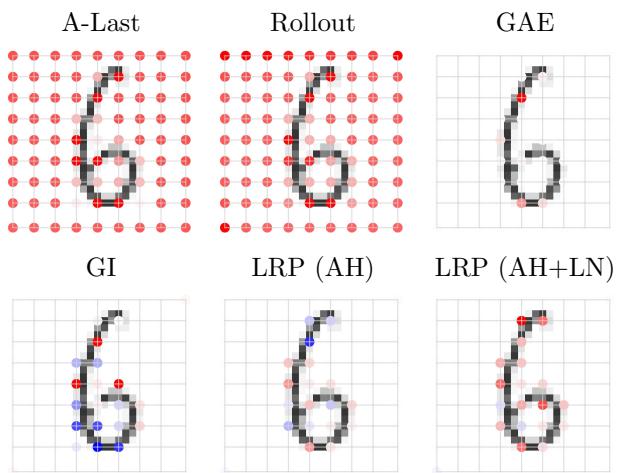


Figure 6: Explanations of MNIST superpixels on a Graphomer model for digit classification. Relevance of a node in the graph is indicated by color and opacity. Red/Blue contribute positively/negatively to the prediction and magnitude is encoded via color opacity.

vide additional examples in Appendix E. Evidently, the proposed improved explanations using LRP (AH) and LRP (AH+LN) indeed highlight the superpixels that contain digit information more reliably than the other approaches. The advantage of conserving information is clearly visible when comparing naive LRP with LRP (AH+LN) where the effect of distorted relevance propagation is apparent. The weaker performance of raw attention-based methods (A-Last, Rollout and GAE) in the perturbation experiments can now be explained by a lack of specificity to the salient digit in favor of attributing more relevance to uninformative background superpixels.

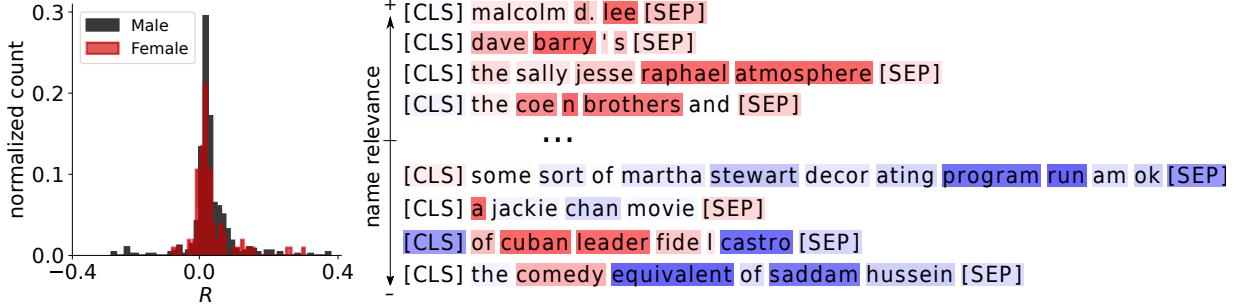


Figure 7: Investigating gender bias in a pre-trained DistillBERT model for sentiment classification on SST-2 movie reviews. Distribution of normalized name occurrences over relevance scores is shown along with sentence samples that contain names that most (top rows)/ least (lower rows) influence a classification towards positive/negative sentiment.

## 6.4 Use Case: Analyzing Bias in Transformers

We now use our method on a popular Transformer architecture, DistillBERT Sanh et al. [2019], to study the detection of systematic bias in machine learning systems through XAI. We download the publicly available checkpoint for sentiment classification on SST-2 from HuggingFace<sup>2</sup> and apply the implementation trick introduced in Section 4. In order to detect such bias, template-based approaches, i.e. “<name> is a successful <job\\_title>”, have been used to test the behavior of the model regarding different systematic relations between, for example, demographics and most likely model predictions Kiritchenko and Mohammad [2018], Prabhakaran et al. [2019], De-Arteaga et al. [2019], Ribeiro et al. [2020], Ousidhoum et al. [2021]. While this is a flexible approach, it involves the risk of producing model inputs that are out of the training distribution and thus can cause unstable predictions.

Instead, we study relevance attribution to bias-sensitive groups of tokens that are of interest. In this example, we explore the possible gender bias in sentiment analysis using the DistillBERT model. For this, we explain the *difference* between positive and negative model outputs for sentiment classification in order to observe which entities and related gender may exhibit a tendency to be more/less relevant to change the classification towards a positive/negative sentiment.

In Figure 7 (left) we first of all observe that there is no consistent bias for female or male names. Overall, there are more male than female names in the dataset, but the distributions of positive/negative sentiment attributed to them are similar. However, we do observe biased model responses towards certain entity categories: After ranking entities based on their assigned relevance from most to least relevant in Figure 7 (right) we observe that common Western male names such as “lee”, “barry” or “coen” can modulate sentiment the strongest towards positive. Interestingly, the first female entity, “sally jesse raphael”, is ranked high because of her typically male

family name “raphael”. At the other extreme, among the names with the strongest negative impact on sentiment, we find a non-Western family name (“chan”) and male political figures (“saddam hussein”, “castro”).

We further note that, because our XAI-based approach disentangles the contribution of individual words from that of other words in the sentence, our approach is more immune to confounders than a naive approach that would simply look at the correlation between name occurrence and predicted sentiment.

## 7 Conclusion

Transformers are a major development in machine learning, with a strong uptake in practical applications, in particular NLP. Consequently, there is a need to obtain transparency regarding the decisions made by these models.

In this work, we have shown that one cannot take for granted that common XAI methods will necessarily continue to work well on Transformer models, as they do on standard deep neural networks. In particular, within the framework of Layer-wise Relevance Propagation, we have shown that GI fails to implement conservation, a common property of attribution techniques. Furthermore, our analysis pinpointed that attention heads and LayerNorms need to be addressed specifically. We also proposed specific and easily to implement LRP rules for these layers.

Experiments demonstrate that our method systematically achieves state-of-the-art results on Transformer and Graphomer models over a broad range of datasets. Lastly, we have showcased our explanation technique on the problem of detecting biases in a sentiment model. Our explanation technique was able to characterize model bias in a detailed manner, without having to generate counterfactual examples and the risk of stepping out of the data manifold. In conclusion, our proposed novel XAI framework for Transformers will ultimately contribute to a safer, more transparent and fair use of state-of-the-art machine learning techniques.

<sup>2</sup><https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english>

## Acknowledgements

KRM was partially supported by the German Ministry for Education and Research under Grant 01IS14013A-E, Grant 01GQ1115, Grant 01GQ0850, as BIFOLD (ref. 01IS18025A and ref. 01IS18037A) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea Government under Grants 2017-0-00451 and 2019-0-00079.

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research, innovation programme (grant ERC CoG 725974).

## References

- S. Abnar and W. H. Zuidema. Quantifying attention flow in transformers. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 4190–4197, 2020.
- M. A. Aleisa, N. Beloff, and M. White. Airm: a new ai recruiting model for the saudi arabia labor market. In K. Arai, editor, *Intelligent Systems Conference (IntelliSys) 2021*, volume 3: 296 of *Lecture Notes in Networks and Systems*, pages 105–124, Cham, September 2021. Springer.
- M. Ancona, E. Ceolini, C. Öztireli, and M. H. Gross. Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 169–191. Springer, 2019.
- L. Arras, J. Arjona-Medina, M. Widrich, G. Montavon, M. Gillhofer, K.-R. Müller, S. Hochreiter, and W. Samek. Explaining and Interpreting LSTMs. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 211–238. Springer, Cham, 2019.
- A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.
- P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. A diagnostic study of explainability techniques for text classification. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3256–3274. Association for Computational Linguistics, 2020.
- S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 2015.
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 4356–4364, Red Hook, NY, USA, 2016. Curran Associates Inc.
- E. Chapuis, P. Colombo, M. Manica, M. Labeau, and C. Clavel. Hierarchical pre-training for sequence labelling in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2636–2648, 2020.
- H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, 2021a.
- H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021b.
- M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. A survey of the state of explainable AI for natural language processing. In *AACL/IJCNLP*, pages 447–459. Association for Computational Linguistics, 2020.
- M. De-Arteaga, A. Romanov, H. Wallach, J. Chayes, C. Borgs, A. Chouldechova, S. Geyik, K. Kenthapadi, and A. T. Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, page 120–128, New York, NY, USA, 2019. Association for Computing Machinery.
- M. Denil, A. Demiraj, and N. de Freitas. Extraction of salient sentences from labelled documents. *ArXiv*, abs/1412.6815, 2014.
- J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186, 2019.

- Y. Ding, Y. Liu, H. Luan, and M. Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1150–1159, 2017.
- A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.
- S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.
- H. Gonen and Y. Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 609–614, 2019.
- R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019.
- R. Hesse, S. Schaub-Meyer, and S. Roth. Fast axiomatic attribution for neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- N. Hollenstein and L. Beinborn. Relative importance in sentence processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 141–150, 2021.
- Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.
- S. Jain and B. C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- S. Kiritchenko and S. Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, 2018.
- S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 4765–4774, 2017.
- D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, volume 33, 2020.
- A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, 2011.
- L. Maziarka, T. Danel, S. Mucha, K. Rataj, J. Tabor, and S. Jastrzkebski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.
- G. Montavon. Gradient-based vs. propagation-based explanations: An axiomatic comparison. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 253–265. Springer International Publishing, 2019.
- G. Montavon, W. Samek, and K. Müller. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73:1–15, 2018.
- F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.
- D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee, and M. Zaharia. Efficient large-scale language model training on GPU clusters using megatron-lm. In *SC ’21: The International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021.
- N. Ousidhoum, X. Zhao, T. Fang, Y. Song, and D.-Y. Yeung. Probing toxic content in large pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4262–4274. Association for Computational Linguistics, 2021.
- P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10772–10781, 2019.

- V. Prabhakaran, B. Hutchinson, and M. Mitchell. Perturbation sensitivity analysis to detect unintended model biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 5739–5744, 2019.
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.
- M. T. Ribeiro, T. S. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proc. Association for Computational Linguistics (ACL)*, pages 4902–4912, 2020.
- W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, editors. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*. Springer, 2019.
- W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.
- V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.
- T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 10.1109/TPAMI.2021.3115452, 2021.
- L. S. Shapley. *A Value for n-Person Games*, pages 307–318. Princeton University Press, 1953.
- A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3145–3153, 2017.
- R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 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, 2013.
- S. Srinivas and F. Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 4126–4135, 2019.
- E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, 2010.
- G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny. Computational modeling of  $\beta$ -secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.
- M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. 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, 2019.
- E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh. AllenNLP interpret: A framework for explaining predictions of NLP models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12, 2019.
- Z. Wu and D. C. Ong. On explaining your explanations of bert: An empirical study with sequence classification. *CoRR*, abs/2101.00196, 2021.
- Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- W. Xiong, J. Wu, H. Wang, V. Kulkarni, M. Yu, X. Guo, S. Chang, and W. Y. Wang. Tweetqa: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, 2019.
- C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, pages 9240–9251, 2019.
- S.-Y. Yoo, Y.-S. Kim, K. Lee, K. Jeong, J. Choi, H. Lee, and Y. S. Choi. Graph-aware transformer: Is attention all graphs need? *ArXiv*, abs/2006.05213, 2020.
- S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim. Graph transformer networks. In *Advances in Neural Information*

*Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 11960–11970, 2019.

J. Zhao, C. Li, Q. Wen, Y. Wang, Y. Liu, H. Sun, X. Xie, and Y. Ye. Gophormer: Ego-graph transformer for node classification. *arXiv preprint arXiv:2110.13094*, 2021.

## A Derivations for Attention Heads

### A.1 Gradient of Softmax

We first consider the softmax function given by:

$$p_{ij} = \frac{\exp(q_{ij})}{\alpha_j} \quad \alpha_j = \sum_i \exp(q_{ij})$$

where  $q_{ij}$  is a shortcut notation for  $q(x_i, x'_j)$ . We can derive its gradient with respect to the tokens  $x_i$  and  $x'_j$ :

$$\begin{aligned} \frac{\partial p_{kj}}{\partial x_i} &= \frac{1}{\alpha_j^2} \left( \exp(q_{kj}) \frac{\partial q_{kj}}{\partial x_i} \alpha_j - \exp(q_{kj}) \exp(q_{ij}) \frac{\partial q_{ij}}{\partial x_i} \right) \\ &= p_{kj} \frac{\partial q_{kj}}{\partial x_i} - p_{kj} p_{ij} \frac{\partial q_{ij}}{\partial x_i} \\ &= p_{kj} [\delta_{ik} - p_{ij}] \frac{\partial q_{ij}}{\partial x_i} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial p_{il}}{\partial x'_j} &= \frac{1}{\alpha_l^2} \left( \exp(q_{il}) \frac{\partial q_{il}}{\partial x'_j} \alpha_l - \exp(q_{il}) \delta_{jl} \sum_k \exp(q_{kl}) \frac{\partial q_{kl}}{\partial x'_j} \right) \\ &= p_{il} \frac{\partial q_{il}}{\partial x'_j} - p_{ij} \delta_{jl} \sum_k p_{kj} \frac{\partial q_{kj}}{\partial x'_j} \\ &= \delta_{jl} p_{ij} \left( \frac{\partial q_{ij}}{\partial x'_j} - \mathbb{E}_j \left[ \frac{\partial q_{ij}}{\partial x'_j} \right] \right) \end{aligned}$$

### A.2 Gradient Propagation Rule for Attention Heads

We recall that the attention head stated in Section 3.1 produces the output

$$y_j = \sum_i x_i p_{ij}.$$

We then state the multivariate chain rule for propagating gradients into tokens of the first sequence and resolve local derivatives, making use in particular of the preliminary derivations of Appendix A.1:

$$\begin{aligned} \frac{\partial f}{\partial x_i} &= \sum_j \left( \frac{\partial y_j}{\partial x_i} + \sum_k \frac{\partial p_{kj}}{\partial x_i} \frac{\partial y_j}{\partial p_{kj}} \right) \frac{\partial f}{\partial y_j} \\ &= \sum_j p_{ij} \frac{\partial f}{\partial y_j} + \sum_{jk} p_{kj} [\delta_{ik} - p_{ij}] \frac{\partial q_{ij}}{\partial x_i} x_k^\top \frac{\partial f}{\partial y_j} \\ &= \sum_j p_{ij} \frac{\partial f}{\partial y_j} + \sum_j p_{ij} \frac{\partial q_{ij}}{\partial x_i} (x_i - \mathbb{E}_j[x])^\top \frac{\partial f}{\partial y_j} \end{aligned}$$

Similarly, we get for the second sequence:

$$\begin{aligned} \frac{\partial f}{\partial x'_j} &= \sum_l \sum_i \frac{\partial p_{il}}{\partial x'_j} \frac{\partial y_l}{\partial p_{il}} \frac{\partial f}{\partial y_l} \\ &= \sum_l \sum_i \delta_{jl} p_{ij} \left( \frac{\partial q_{ij}}{\partial x'_j} - \mathbb{E}_j \left[ \frac{\partial q_{ij}}{\partial x'_j} \right] \right) x_i^\top \frac{\partial f}{\partial y_l} \\ &= \sum_i p_{ij} \left( \frac{\partial q_{ij}}{\partial x'_j} - \mathbb{E}_j \left[ \frac{\partial q_{ij}}{\partial x'_j} \right] \right) x_i^\top \frac{\partial f}{\partial y_j} \\ &= \mathbb{E}_j \left[ \left( \frac{\partial q_{ij}}{\partial x'_j} - \mathbb{E}_j \left[ \frac{\partial q_{ij}}{\partial x'_j} \right] \right) x_i^\top \frac{\partial f}{\partial y_j} \right] \end{aligned}$$

### A.3 Relevance Conservation

Computing the relevance for the first sequence

$$\begin{aligned}\sum_i \mathcal{R}(x_i) &= \sum_i x_i^\top \frac{\partial f}{\partial x_i} \\ &= \sum_j \mathcal{R}(y_j) + \sum_{ij} p_{ij} x_i^\top \frac{\partial q_{ij}}{\partial x_i} (x_i - \mathbb{E}_j[x])^\top \frac{\partial f}{\partial y_j} \\ &= \sum_j \mathcal{R}(y_j) + \sum_j \mathbb{E}_j [q_{:j} \cdot (x - \mathbb{E}_j[x])^\top] \frac{\partial f}{\partial y_j}\end{aligned}$$

and for the second sequence

$$\begin{aligned}\sum_j \mathcal{R}(x'_j) &= \sum_j x'_j^\top \frac{\partial f}{\partial x'_j} \\ &= \sum_j x'_j^\top \mathbb{E}_j \left[ \left( \frac{\partial q_{:j}}{\partial x'_j} - \mathbb{E}_j \left[ \frac{\partial q_{:j}}{\partial x'_j} \right] \right) x^\top \right] \frac{\partial f}{\partial y_j} \\ &= \sum_j \mathbb{E}_j [(q_{:j} - \mathbb{E}_j[q_{:j}]) x^\top] \frac{\partial f}{\partial y_j},\end{aligned}$$

and summing the two scores, we obtain:

$$\begin{aligned}\sum_i \mathcal{R}(x_i) + \sum_j \mathcal{R}(x'_j) &= \sum_j \mathcal{R}(y_j) + \sum_j \left( \mathbb{E}_j [(q_{:j} - \mathbb{E}_j[q_{:j}]) x^\top] + \mathbb{E}_j [q_{:j} \cdot (x - \mathbb{E}_j[x])^\top] \right) \frac{\partial f}{\partial y_j}\end{aligned}$$

Making the further assumption that the first sequence of tokens and the softmax input both have expected value zero, we obtain the simplified form:

$$\begin{aligned}\sum_i \mathcal{R}(x_i) + \sum_j \mathcal{R}(x'_j) &= \sum_j \mathcal{R}(y_j) + \sum_j 2 \text{Cov}_j(q_{:j}, x)^\top \frac{\partial f}{\partial y_j}\end{aligned}$$

## B Derivations for LayerNorm

The core part of LayerNorm can be decomposed into two parts:

$$\tilde{x}_i = x_i - \mathbb{E}[x] \tag{10}$$

$$y_i = \frac{\tilde{x}_i}{\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]}} \tag{11}$$

i.e. a centering followed by a rescaling, where  $\mathbb{E}[\cdot]$  is computed over a uniform distribution, i.e.  $\mathbb{E}[x] = \frac{1}{N} \sum_i x_i$ .

### B.1 Centering step

We first analyze GI relevance conservation in the centering step. For this, we make use of Eq. (4) of the main paper, and perform a summation over input variables on both sides:

$$\sum_i \mathcal{R}(x_i) = \sum_i \sum_j \frac{\partial \tilde{x}_j}{\partial x_i} \frac{x_i}{\tilde{x}_j} \mathcal{R}(\tilde{x}_j) \tag{12}$$

Observing that  $\tilde{x}_j(x)$  is a homogeneous function of degree one, and that consequently  $\sum_i x_i \frac{\partial \tilde{x}_j}{\partial x_i} = \tilde{x}_j$  for all  $j$ , Eq. (12) reduces to

$$\sum_i \mathcal{R}(x_i) = \sum_i \mathcal{R}(\tilde{x}_i) \tag{13}$$

## B.2 Rescaling step

We now analyze conservation in the rescaling part of LayerNorm. The local derivative is given by:

$$\begin{aligned}\frac{\partial y_j}{\partial \tilde{x}_i} &= \frac{\delta_{ij}}{\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]}} - \frac{\tilde{x}_j \frac{\partial}{\partial \tilde{x}_i} (\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]})}{(\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]})^2} \\ &= \frac{\delta_{ij}}{\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]}} - \frac{\frac{1}{N} \tilde{x}_i \tilde{x}_j}{(\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]})^3}\end{aligned}$$

We can now analyze conservation by applying Eq. (4) of the paper:

$$\begin{aligned}\sum_i \mathcal{R}(\tilde{x}_i) &= \sum_i \sum_j \frac{\partial y_j}{\partial \tilde{x}_i} \frac{\tilde{x}_i}{y_j} \mathcal{R}(y_j) \\ &= \sum_i \sum_j \left( \frac{\delta_{ij}}{\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]}} - \frac{\frac{1}{N} \tilde{x}_i \tilde{x}_j}{(\sqrt{\epsilon + \mathbb{E}[\tilde{x}^2]})^3} \right) \frac{\tilde{x}_i}{y_j} \mathcal{R}(y_j) \\ &= \sum_i \sum_j (\delta_{ij} y_i - \frac{1}{N} y_i y_i y_j) \cdot \frac{1}{y_j} \mathcal{R}(y_j) \\ &= \sum_j (1 - \mathbb{E}[y_j^2]) \mathcal{R}(y_j) \\ &= \left( 1 - \frac{\mathbb{E}[\tilde{x}^2]}{\epsilon + \mathbb{E}[\tilde{x}^2]} \right) \sum_j \mathcal{R}(y_j)\end{aligned}$$

Hence, after combining with the centering step, we obtain the conservation equation:

$$\sum_i \mathcal{R}(x_i) = \left( 1 - \frac{\text{Var}[x]}{\epsilon + \text{Var}[x]} \right) \sum_i \mathcal{R}(y_i) \quad (14)$$

where conservation holds only approximately for large values of  $\epsilon$ .

## C Implementation Details

For SST-2 and IMDB sentiment classification, the embeddings module and the tokenizer are initialized from pre-trained BERT-Transformers (textattack/bert-base-uncased-{{sst-2/imdb}}). We report the following test accuracies after optimization: 81.3% for SST-2, 88.1% on IMDB. For training, we use batchsizes of  $bs = 32$  and optimize the model parameters using the AdamW optimizer with a learning rate of  $lr = 2e - 5$  for a maximal number of  $T = 20$  epochs or until early stopping for decreasing validation performance is reached. For the other NLP tasks we follow the same settings, except that we initialize the embedding model and the tokenizer from a HuggingFace pre-trained BERT-Transformer (bert-base-uncased).

For the BACE and MNIST-Superpixels datasets, the recent Graphomer model is used. We train a 2-layer graphomer model with a batchsize of  $bs = 64$  and AdamW as an optimizer with  $lr = 2e - 4$ . The model is trained for 1000 epochs for the BACE dataset and 10 epochs for MNIST-Superpixels, achieving an accuracy of 97.2% for MNIST, and a ROC-ACC of 83.5% for BACE.

## D Additional Qualitative Results on SST-2

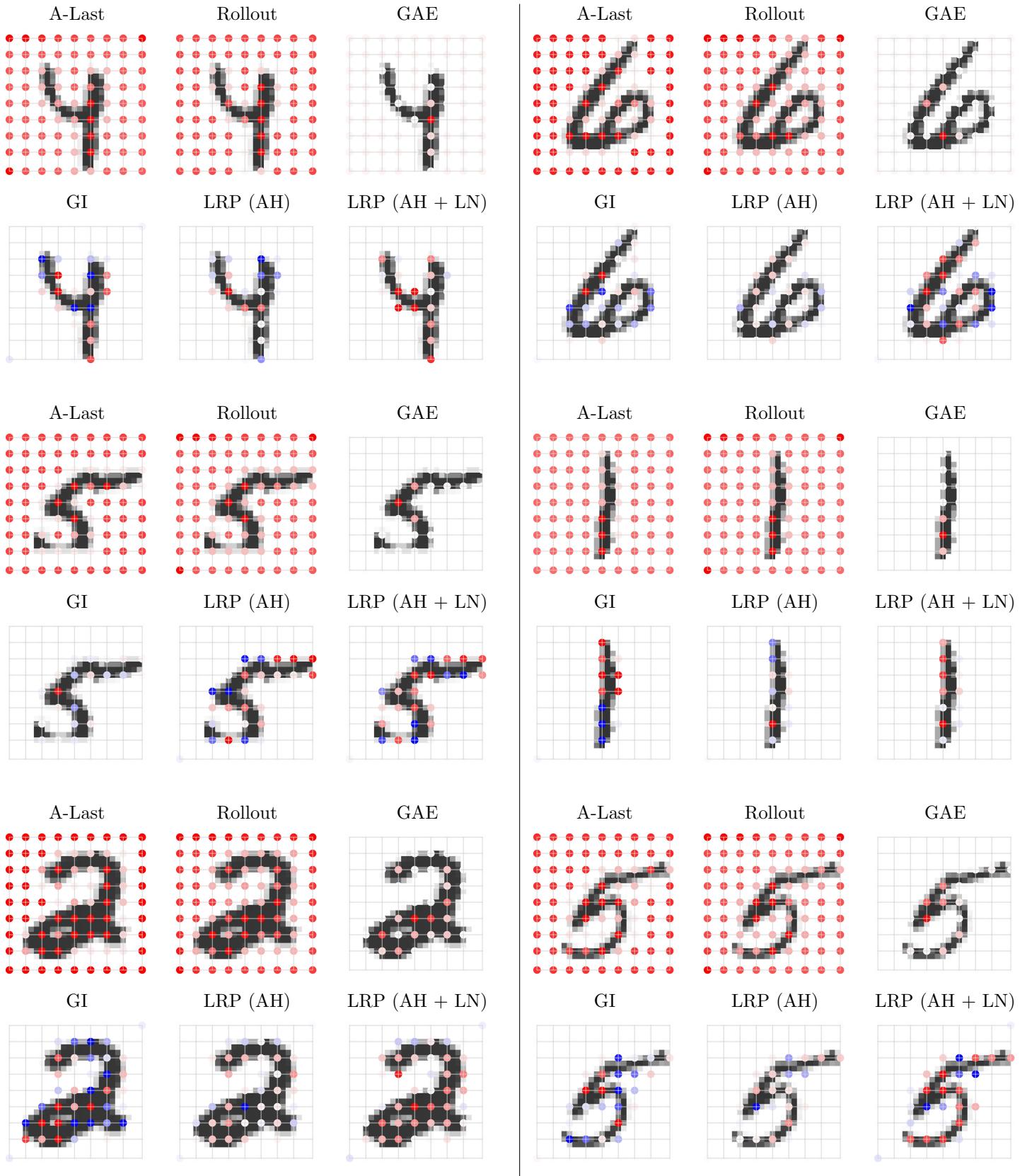
A-last	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	A-last	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
A-Flow	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	A-Flow	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
Rollout	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	Rollout	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
GAE	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	GAE	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
GI	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	GI	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
LRP (AH)	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	LRP (AH)	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]
LRP (AH+LN)	[CLS] a <b>gorgeous, witty, seductive movie.</b> [SEP]	LRP (AH+LN)	[CLS] and the <b>lesson, in the end, is nothing new.</b> [SEP]

A-last	[CLS] this movie is <b>madden #ing.</b> [SEP]	A-last	[CLS] i ' m just <b>too bored to care.</b> [SEP]
A-Flow	[CLS] this movie is <b>madden #ing.</b> [SEP]	A-Flow	[CLS] i ' m just <b>too bored to care.</b> [SEP]
Rollout	[CLS] this movie is <b>madden #ing.</b> [SEP]	Rollout	[CLS] i ' m just <b>too bored to care.</b> [SEP]
GAE	[CLS] this movie is <b>madden #ing.</b> [SEP]	GAE	[CLS] i ' m just <b>too bored to care.</b> [SEP]
GI	[CLS] this movie is <b>madden #ing.</b> [SEP]	GI	[CLS] i ' m just <b>too bored to care.</b> [SEP]
LRP (AH)	[CLS] this movie is <b>madden #ing.</b> [SEP]	LRP (AH)	[CLS] i ' m just <b>too bored to care.</b> [SEP]
LRP (AH+LN)	[CLS] this movie is <b>madden #ing.</b> [SEP]	LRP (AH+LN)	[CLS] i ' m just <b>too bored to care.</b> [SEP]

A-last	[CLS] just one <b>bad idea after another.</b> [SEP]	A-last	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
A-Flow	[CLS] just one <b>bad idea after another.</b> [SEP]	A-Flow	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
Rollout	[CLS] just one <b>bad idea after another.</b> [SEP]	Rollout	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
GAE	[CLS] just one <b>bad idea after another.</b> [SEP]	GAE	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
GI	[CLS] just one <b>bad idea after another.</b> [SEP]	GI	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
LRP (AH)	[CLS] just one <b>bad idea after another.</b> [SEP]	LRP (AH)	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]
LRP (AH+LN)	[CLS] just one <b>bad idea after another.</b> [SEP]	LRP (AH+LN)	[CLS] a sequence of <b>ridiculous shoot - ' em - up scenes.</b> [SEP]

A-last	[CLS] it <b>treats women like idiots.</b> [SEP]	A-last	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
A-Flow	[CLS] it <b>treats women like idiots.</b> [SEP]	A-Flow	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
Rollout	[CLS] it <b>treats women like idiots.</b> [SEP]	Rollout	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
GAE	[CLS] it treats women like <b>idiots.</b> [SEP]	GAE	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
GI	[CLS] it <b>treats women like idiots.</b> [SEP]	GI	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
LRP (AH)	[CLS] it treats women like <b>idiots.</b> [SEP]	LRP (AH)	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]
LRP (AH+LN)	[CLS] it treats women like <b>idiots.</b> [SEP]	LRP (AH+LN)	[CLS] there is <b>no pleasure in watching a child suffer.</b> [SEP]

## E Additional Qualitative Results on MNIST Superpixels



## F Perturbation Experiments

