AttnLRP: Attention-Aware Layer-Wise Relevance Propagation for Transformers

Reduan Achtibat [1] **Sayed Mohammad Vakilzadeh Hatefi** [1] **Maximilian Dreyer** [1] **Aakriti Jain** [1]
Thomas Wiegand [1 2 3] **Sebastian Lapuschkin** [1] **Wojciech Samek** [1 2 3]

Abstract

Large Language Models are prone to biased predictions and hallucinations, underlining the paramount importance of understanding their model-internal reasoning process. However, achieving faithful attributions for the entirety of a black-box transformer model and maintaining computational efficiency is an unsolved challenge. By extending the Layer-wise Relevance Propagation attribution method to handle attention layers, we address these challenges effectively. While partial solutions exist, our method is the first to faithfully and holistically attribute not only input but also latent representations of transformer models with the computational efficiency similar to a single backward pass. Through extensive evaluations against existing methods on LLaMa 2, Mixtral 8x7b, Flan-T5 and vision transformer architectures, we demonstrate that our proposed approach surpasses alternative methods in terms of faithfulness and enables the understanding of latent representations, opening up the door for concept-based explanations. We provide an LRP library at https://github.com/rachtibat/LRP-eXplains-Transformers.

1. Introduction

The attention mechanism (Vaswani et al., 2017) became an essential component of large transformers due to its unique ability to handle multimodality and to scale to billions of training samples. While these models demonstrate impressive performance in text and image generation, they are prone to biased predictions and hallucinations (Huang et al., 2023), which hamper their widespread adoption.

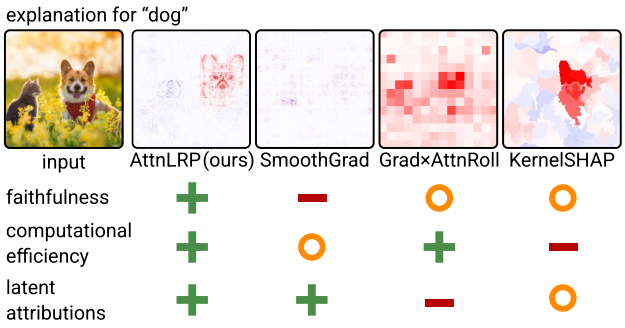

Figure 1. By optimizing LRP for transformer-based architectures, our LRP variant outperforms other state-of-the-art methods in terms of explanation faithfulness and computational efficiency. We further are able to explain latent neurons inside and outside the attention module, allowing us to interact with the model. A more detailed discussion on the differences between AttnLRP and other LRP variants can be found in Appendix A.2.2. Heatmaps for other methods are illustrated in Appendix Figure B.6. Legend: highly (+), semi- (○), not (−) suited. Credit: Nataba/iStock.

To overcome these limitations, it is crucial to understand the latent reasoning process of transformer models. Researchers started using the attention mechanism of transformers as a means to understand how input tokens interact with each other. Attention maps contain rich information about the data distribution (Clark et al., 2019; Caron et al., 2021), even allowing for image data segmentation. However, attention, by itself, is inadequate for comprehending the full spectrum of model behavior (Wiegreffe and Pinter, 2019). Similar to latent activations, attention is not class-specific and solely provides an explanation for the softmax output (in attention layers) while disregarding other model components. Recent works (Geva et al., 2021; Dai et al., 2022) have in fact discovered that factual knowledge in Large Language Models (LLMs) is stored in Feed-Forward Network (FFN) neurons, separate from attention layers. Further, attention-based attribution methods such as rollout (Abnar and Zuidema, 2020; Chefer et al., 2021a) result in checkerboard artifacts, as visible in Figure 1 for

[1]Fraunhofer Heinrich-Hertz-Institute, 10587 Berlin, Germany [2]Technische Universität Berlin, 10587 Berlin, Germany [3]BIFOLD – Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany. Correspondence to: Wojciech Samek <wojciech.samek@hhi.fraunhofer.de>, Sebastian Lapuschkin <sebastian.lapuschkin@hhi.fraunhofer.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

a Vision Transformer (ViT). Researchers thus have turned to model-agnostic approaches that aim to provide a holistic explanation of the model's behavior (Miglani et al., 2023), including, *e.g.*, perturbation and gradient-based methods.

Methods based on feature perturbation require excessive amounts of compute time (and energy), and in order to access latent attributions they require performing perturbations at each layer separately, resulting in further exponential cost increase. This makes their application economically infeasible, especially for large architectures. In contrast, gradient-based methods benefit from the chain-rule in automatic differentiation and can produce latent attributions for all layers in a single backward pass. While prominent gradient-based methods, *e.g.* Input $\times$ Gradient (Simonyan et al., 2014), are highly efficient, they suffer from noisy gradients and low faithfulness, as evaluated in Section 4.1.

Another option is to take advantage of the versatility of *rule-based* backpropagation methods, such as Layer-wise Relevance Propagation (LRP). These methods allow for the customization of propagation rules to accommodate novel operations, allowing for more faithful explanations and requiring only a single backward pass. As thoroughly discussed in Appendix A.2.2, all previous attempts to apply LRP to transformers reused standard LRP rules (Ding et al., 2017; Voita et al., 2021; Chefer et al., 2021b; Ali et al., 2022). However, transformer architectures include several functions for which standard LRP rules do not adequately apply, such as softmax, bi-linear (matrix) multiplication (*e.g.* query-key multiplication) and layer normalization. Additionally, the routing networks in Mixture of Experts (MoE) models (Fedus et al., 2022) present notable challenges due to their combination of these functions. As a result, other previous attempts result in either numerical instabilities or low faithfulness.

Our method represents a significant breakthrough in handling the attribution problem in transformer architectures by enabling an accurate attribution flow through non-linear model components outperforming other existing methods (including perturbation) by a large margin.

Contributions In this work, we introduce AttnLRP, an extension of LRP within the Deep Taylor Decomposition framework (Montavon et al., 2017), with the particular requirements necessary for attributing non-linear transformer components accurately. AttnLRP allows explaining transformer-based models with high faithfulness and efficiency, while also allowing attribution of latent neurons and providing insights into their role in the generation process (see Figure 2).

1. We derive novel efficient and faithful LRP attribution rules for non-linear attention within the Deep Taylor Decomposition framework, demonstrating their superiority over the state-of-the-art and successfully tackling the noise problem in ViTs.
2. We illustrate how to gain insights into an LLM generation process by identifying relevant neurons and explaining their encodings.
3. We provide an efficient and ready-to-use open source implementation of AttnLRP for transformers.

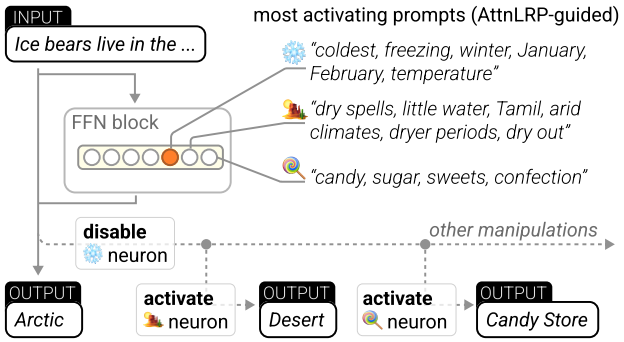

Figure 2. AttnLRP combined with ActMax allows to identify relevant neurons and gain insights into their encodings. This allows one to manipulate the latent representations and, *e.g.*, to change the output "Arctic" (by disabling the corresponding neuron) to "Desert" or "Candy Store" (by activating the respective neurons). See also Section 4.3.

2. Related Work

We present an overview of related work for various model-agnostic and transformer-specialized attribution methods.

2.1. Perturbation & Local Surrogates

In perturbation analysis, such as occlusion-based attribution (Zeiler and Fergus, 2014) or SHAP (Lundberg and Lee, 2017), the input features are repeatedly perturbed while the effect on the model output is measured (Fong and Vedaldi, 2017). AtMan (Deb et al., 2023) is specifically adapted to the transformer architecture, where tokens are not suppressed in the input space, but rather in the latent attention weights.

Interpretable local surrogates, on the other hand, replace complex black-box models with simpler linear models that locally approximate the model function being explained. Since the surrogate has low complexity, interpretability is facilitated. Prominent methods include LIME (Ribeiro et al., 2016) and LORE (Guidotti et al., 2018).

While these approaches are model-agnostic and memory efficient, they have a high computational cost in terms of forward passes. Furthermore, explanations generated on surrogate models cannot explain the hidden representations

of the original model. Finally, latent attributions wrt. the prediction must be computed for each layer separately, increasing the computational cost further.

2.2. Attention-based

These methods take advantage of the attention mechanism in transformer models. Although attention maps capture parts of the data distribution, they lack class specificity and do not provide a meaningful interpretation of the final prediction (Wiegreffe and Pinter, 2019). Attention Rollout (Abnar and Zuidema, 2020) attempts to address the issue by sequentially connecting attention maps of all layers. However, the resulting attributions are still not specific to individual outputs and exhibit substantial noise. Hence, (Gildenblat, 2023) has found that reducing noise in attention rollout can be achieved by filtering out excessively strong outlier activations.

To enable class-specificity, the work of (Chefer et al., 2021b) proposed a novel rollout procedure wherein the attention's activation is mean-weighted using a combination of the gradient and LRP-inspired relevances. It is important to note that this approach yields an approximation of the mean squared relevance value, which diverges from the originally defined notion of "relevance" or "importance" of additive explanatory models such as SHAP (Lundberg and Lee, 2017) or LRP (Bach et al., 2015). Subsequent empirical observations by (Chefer et al., 2021a) revealed that an omission of LRP-inspired relevances and a sole reliance on a positive mean-weighting of the attention's activation with the gradient improved the faithfulness inside cross-attention layers. Though, this approach can only attribute positively and does not consider counteracting evidence.

Attention-rollout based approaches, while offering advantages in terms of computational efficiency and conceptual simplicity, have significant drawbacks. Primarily, they suffer from a limited resolution in the input attribution maps, resulting in undesirable checkerboard artifacts cf. Figure 1. Moreover, they are unable to attribute hidden latent features beyond the softmax output. Consequently, these approaches only provide explanations for a fraction of the model, thereby compromising the fidelity and limiting the feasibility of explanations within the hidden space.

2.3. Backpropagation-based

Input $\times$ Gradient (Simonyan et al., 2014) linearizes the model by utilizing the gradient. However, this approach is vulnerable to gradient shattering (Balduzzi et al., 2017; Dombrowski et al., 2022), leading to noisy attributions in deep models. Consequently, several works aim to denoise these attributions. SmoothGrad (Smilkov et al., 2017) and Integrated Gradients (Sundararajan et al., 2017) have attempted to address this issue but have been unsuccessful in the case of large transformers, as demonstrated in the experiments in Section 4.1. (Chefer et al., 2021b) adapted Grad-CAM (Selvaraju et al., 2017) to transformer models by weighting the last attention map with the gradient.

Modified backpropagation methods, such as LRP (Bach et al., 2015), decompose individual layer functions instead of linearizing the entire model. They modify the gradient to produce more reliable attributions (Arras et al., 2022). The work (Ding et al., 2017) was the first to apply standard LRP on non-linear attention layers, while (Voita et al., 2021) proposed an improved variant building upon the Deep Taylor Decomposition framework. Nonetheless, both variants can lead to numerical instabilities in attributing the softmax function and do not fulfill the conservation property (3) in matrix multiplication. (Ali et al., 2022) considerably improved attributions by recognizing that standard LRP rules were not suitable for these operations and proposed to exclude softmax and normalization operations from the computational graph by stopping the relevance (gradient) flow through them. However, it does not resolve the fundamental challenge of optimally applying LRP to non-linear operations. In Appendix A.2.2, we provide a comprehensive analysis about different LRP-variants.

3. Attention-Aware LRP for Transformers

First, we motivate LRP in the framework of additive explanatory models. Then, we generalize the design of new rules for non-linear operations. Finally, we apply our methodology successively on each operation utilized in a transformer model to derive efficient and faithful rules.

3.1. Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) (Bach et al., 2015; Montavon et al., 2019) belongs to the family of additive explanatory models, which includes the well-known Shapley (Lundberg and Lee, 2017), Gradient $\times$ Input (Simonyan et al., 2014) and DeepLIFT (Shrikumar et al., 2017) methods.

The underlying assumption of such models is that a function f_j with N input features $\mathbf{x} = \{x_i\}_{i=1}^N$ can be decomposed into individual contributions of single input variables $R_{i\leftarrow j}$ (called "relevances"). Here, $R_{i\leftarrow j}$ denotes the amount of output j that is attributable to input i, which, when added together, equals (or is proportional to) the original function value. Mathematically, this can be written as:

$$f_j(\mathbf{x}) \propto R_j = \sum_i^N R_{i\leftarrow j} \tag{1}$$

If an input i is connected to several outputs j, *e.g.*, a multidimensional function $\mathbf{f}$, the contributions of each output j

are losslessly aggregating together.

$$R_i = \sum_j R_{i \leftarrow j}. \tag{2}$$

This provides us with "importance values" for the input variables, which reveal their direct contribution to the final prediction. Unlike other methods, LRP treats a neural network as a layered directed acyclic graph, where each neuron j in layer l is modeled as a function node f_j^l that is individually decomposed according to Equation (1). Beginning at the model output L, the initial relevance value $R_j^L \propto f_j^L$ is successively distributed to its prior network neurons one layer at a time. Hence, LRP follows the flow of activations computed during the forward pass through the model in the *opposite* direction, from output f^L back to input layer f^1.

This decomposition characteristic of LRP gives rise to the important *conservation property*:

$$R^{l-1} = \sum_i R_i^{l-1} = \sum_{i,j} R_{i \leftarrow j}^{(l-1,l)} = \sum_j R_j^l = R^l \tag{3}$$

ensuring that the sum of all relevance values in each layer remains constant. This property allows for meaningful attribution, as the scale of each relevance value can be related to the original function output f^L.

3.1.1. Decomposition through Linearization

To design a faithful attribution method, the challenge lies in identifying a meaningful distribution rule $R_{i \leftarrow j}$. Possible solutions encompass all decompositions that adhere to the conservation property (3). However, for a decomposition to be considered *faithful*, it should approximate the characteristics of the original function as closely as possible.

In this paper, we take advantage of the Deep Taylor Decomposition framework (Montavon et al., 2017) to locally linearize and decompose neural network operations into independent contributions. As a special case, we further establish the relationship between one derived rule and the Shapley Values framework in Section 3.3.2.

We start by computing a first-order Taylor expansion at a reference point $\tilde{\mathbf{x}}$. For the purpose of simplifying the equation, we assume that the reference point $\tilde{\mathbf{x}}$ is constant:

$$\begin{aligned} f_j(\mathbf{x}) &= f_j(\tilde{\mathbf{x}}) + \sum_i \mathbf{J}_{ji}(\tilde{\mathbf{x}})\,(x_i - \tilde{x}_i) + \mathcal{O}(|\mathbf{x} - \tilde{\mathbf{x}}|^2) \quad (4) \\ &= \sum_i \mathbf{J}_{ji}\, x_i + \underbrace{f_j(\tilde{\mathbf{x}}) - \sum_i \mathbf{J}_{ji}\, \tilde{x}_i}_{\text{bias } \tilde{b}_j} + \mathcal{O}(|\mathbf{x} - \tilde{\mathbf{x}}|^2) \end{aligned}$$

where $\mathcal{O}$ is the approximation error in Big-$\mathcal{O}$ notation and the Jacobian $\mathbf{J}$ is evaluated at reference point $\tilde{\mathbf{x}}$, that is in the following omitted for brevity[1]. The bias term represents the constant portion of the function and the approximation error that cannot be directly attributed to the input variables.

We substitute the layer function with its first-order expansion and assert its proportionality to a relevance value R_j following Equation (1) through multiplication with a constant factor $c \in \mathbb{R}$ with $f_j(\mathbf{x}) \neq 0$.

$$R_j = f_j(\mathbf{x})\, c = \sum_i \underbrace{\mathbf{J}_{ji}\, x_i \frac{R_j}{f_j(\mathbf{x})}}_{R_{i \leftarrow j}} + \underbrace{\tilde{b}_j \frac{R_j}{f_j(\mathbf{x})}}_{R_{b \leftarrow j}}$$

Comparing with Equation (1), we identify $R_{i \leftarrow j}$ as the relevance assigned to the input variables and $R_{b \leftarrow j}$ as the relevance assigned to the bias term. Hence, the bias term absorbs a portion of the relevance R_j that is not allocated to the input variables. This technically violates the conservation property (3), as only $R_{i \leftarrow j}$ is further distributed to prior layers reducing the amount of relevance per distribution step. However, (Bach et al., 2015) treats bias terms as additional hidden neurons (with an activation value of one and a weight that equals the bias value, connected to the output) including them into the conservation property (3). Consequently, we regard this relevance as preserved, rather than lost. Alternatively, to strictly enforce conservation, the absorbed relevance score of the bias term can be distributed equally among the input variables, or the bias term can be excluded completely, as explained in Appendix A.2.1.

To obtain a propagation rule for the input variables, we apply Equation (2) without the bias term. In addition, we insert a stabilizing factor $\varepsilon \ll |f_j(\mathbf{x})| \in \mathbb{R}^+$ with the sign of $f_j(\mathbf{x})$ to allow for the case $f_j(\mathbf{x}) = 0$:

$$R_i = \sum_j R_{i \leftarrow j} = \sum_j \mathbf{J}_{ji}\, x_i \frac{R_j}{f_j(\mathbf{x}) + \varepsilon\, \text{sign}(f_j(\mathbf{x}))} \tag{5}$$

In the following, $\text{sign}(f_j(\mathbf{x})) \in \{-1, 1\}$ is omitted for brevity. Note, that ε acts as bias term and absorbs a negligible amount of the relevance.

To benefit from GPU parallelization, this formula can be written in matrix form:

$$\Rightarrow \mathbf{R}^{l-1} = \mathbf{x} \odot \mathbf{J}^\top \cdot \mathbf{R}^l \oslash (\mathbf{f}(\mathbf{x}) + \varepsilon)$$

where $\odot$ denotes the Hadamard product, $\oslash$ element-wise division and $\mathbf{R}^l$ a relevance vector at layer l. This formula can be efficiently implemented in automatic differentiation libraries, such as PyTorch (Paszke et al., 2019). Compared to a basic backward pass, we have additional computational complexity for the element-wise operations.

[1] if $\tilde{\mathbf{x}} = \mathbf{x}$, this is equivalent to Gradient × Input. We have taken the DTD perspective to highlight the bias term, which is important for the upcoming discussion.

3.2. Attributing the Multilayer Perceptron

Commonly a Multilayer Perceptron consists of a linear layer with a (component-wise) non-linearity producing input activations for the succeeding layer(s):

$$z_j = \sum_i \mathbf{W}_{ji}\, x_i + b_j \tag{6}$$

$$a_j = \sigma(z_j) \tag{7}$$

where $\mathbf{W}_{ji}$ are the weight parameters and σ constitutes a (component-wise) non-linearity.

3.2.1. The ε- and γ-LRP rule

Linearizing linear layers (6) at any point $\mathbf{x} \in \mathbb{R}^N$ results in the fundamental ε-LRP (Bach et al., 2015) rule

$$R_i^{l-1} = \sum_j \mathbf{W}_{ji} x_i \frac{R_j^l}{z_j(\mathbf{x}) + \varepsilon} \tag{8}$$

The bias b_j of Equation (6) and ε absorb a portion of the relevance. The proof is omitted for brevity. We employ the ε-LRP rule on all linear layers, unless specified otherwise.

In models with many layers, the gradient of a layer can cause noisy attributions due to the gradient shattering effect (Balduzzi et al., 2017; Dombrowski et al., 2022). To mitigate this noise, it is best practice to use the γ-LRP rule (Montavon et al., 2019), an extension to improve the signal-to-noise ratio. We have observed that this effect is significantly pronounced in ViTs while LLMs lack visible noise. Therefore, we only apply the γ-LRP rule to linear layers in ViTs. For more details, please refer to Appendix A.2.3.

3.2.2. Handling Element-wise Non-Linearities

Since element-wise non-linearities have only a single input and output variable, the decomposition of Equation (1) *is* the operation itself. Therefore, the entire incoming relevance R_j^l can only be assigned to the single input variable.

$$R_i^{l-1} = R_i^l \tag{9}$$

The identity rule (9) is applied to all element-wise operations with a single input and single output variable.

3.3. Attributing Non-linear Attention

The heart of the transformer architecture (Vaswani et al., 2017) is non-linear attention

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^\top}{\sqrt{d_k}}\right) \tag{10}$$

$$\mathbf{O} = \mathbf{A} \cdot \mathbf{V} \tag{11}$$

$$\text{softmax}_j(\mathbf{x}) = \frac{e^{x_j}}{\sum_k e^{x_k}} \tag{12}$$

where $(\cdot)$ denotes matrix multiplication, $\mathbf{K} \in \mathbb{R}^{b \times s_k \times d_k}$ is the key matrix, $\mathbf{Q} \in \mathbb{R}^{b \times s_q \times d_k}$ is the queries matrix, $\mathbf{V} \in \mathbb{R}^{b \times s_k \times d_v}$ the values matrix, and $\mathbf{O} \in \mathbb{R}^{b \times s_k \times d_v}$ is the final output of the attention mechanism. b is the batch dimension including the number of heads, and d_k, d_v indicate the embedding dimensions, and s_q, s_k are the number of query and key/value tokens.

First and foremost, the softmax function is highly non-linear. In addition, the matrix multiplication is bilinear, *i.e.*, linear in both of its input variables. In the following, we will derive relevance propagation rules for each of these operations, taking into account considerations of efficiency.

3.3.1. Handling the Softmax Non-Linearity

In Section 3.1.1, we present a generalized approach to linearization that incorporates bias terms, allowing for the absorption of a portion of the relevance. However, (Ali et al., 2022) advocates for a strict adherence to the conservation property (3) and argues that a linear decomposition of a *non-linear* function should typically exclude a bias term. While we see the virtue of this approach for operations such as RMSNorm (Zhang and Sennrich, 2019) or matrix multiplication, where $f(0) = 0$, we contend that a linearization of the softmax function should inherently incorporate a bias term. This is due to the fact that even when the input is zero, the softmax function yields a value of $\frac{1}{N}$ (where N represents the dimension of the inputs) which is analogous to a virtual bias term.

Proposition 3.1 *Decomposing the softmax function by a Taylor decomposition (4) at reference point* $\boldsymbol{x}$ *yields the following relevance propagation rule:*

$$R_i^{l-1} = x_i(R_i^l - s_i \sum_j R_j^l) \tag{13}$$

where s_i denotes the i-th output of the softmax function. The hidden bias term, which contains the approximation error, consequently absorbs a portion of the relevance.

The proof can be found in Appendix A.3.1. In Appendix A.2.4, we explore the implications of vanishing gradients and temperature scaling on attributing the softmax function, which is important when attributing softmax outside of the attention mechanism, *e.g.* at the classification output. Note, that the works (Ding et al., 2017; Voita et al., 2021; Chefer et al., 2021b; Ali et al., 2022) propose to handle the bias term differently to strictly enforce the conservation property (3). Most variants can lead to severe numerical instabilities as discussed in Appendix A.2.1 and seen empirically in our preliminary experiments.

3.3.2. Handling Matrix-Multiplication

Since $f(0,0) = 0$ holds, it is desirable to decompose the matrix multiplication without a bias term. To achieve this, we break down the matrix multiplication into an affine operation involving summation and a bi-linear part involving element-wise multiplication.

$$\mathbf{O}_{jp} = \sum_i \underbrace{\mathbf{A}_{ji}\mathbf{V}_{ip}}_{\text{bi-linear part}}$$

The summation already provides a decomposition in the form of Equation (1), and we only need to decompose the individual summands $\mathbf{A}_{ji}\mathbf{V}_{ip}$.

Proposition 3.2 *Decomposing element-wise multiplication with N input variables of the form*

$$f_j(\boldsymbol{x}) = \prod_i^N x_{ji}$$

by Shapley (with baseline zero) or Taylor decomposition (4) at reference point $\boldsymbol{x}$ (without bias or distributing the bias uniformely) yields the following uniform relevance propagation rule:

$$R_{ji}^{l-1} = \frac{1}{N} R_j^l \,. \tag{14}$$

The proof can be found in Appendix A.3.2. Consequently, the combined rule can be effectively computed using:

Proposition 3.3 *Decomposing matrix multiplication with a sequential application of the ε-rule (8) and the uniform rule (14) on the summands yields the following relevance propagation rule for $\boldsymbol{A}_{ji}$:*

$$R_{ji}^{l-1} = \sum_p \mathbf{A}_{ji}\mathbf{V}_{ip} \frac{R_{jp}^l}{2\,\mathbf{O}_{jp} + \varepsilon} \tag{15}$$

There is no bias term absorbing relevance, whereas ε absorbs a negligible quantity. For $\mathbf{V}_{ip}$, we sum over the j indices. The proof can be found in Appendix A.3.3. By employing this rule, we maintain strict adherence to the conservation property (3), as explained in Appendix A.3.5.

3.3.3. Handling Normalization Layers

Commonly used normalization layers in Transformers include LayerNorm (Ba et al., 2016) and RMSNorm (Zhang and Sennrich, 2019). These layers apply affine transformations and non-linear normalization sequentially.

$$\text{LayerNorm}(\mathbf{x}) = \frac{x_j - \mathbb{E}[\mathbf{x}]}{\sqrt{\text{Var}[\mathbf{x}] + \varepsilon}}\gamma_j + \beta_j \tag{16}$$

$$\text{RMSNorm}(\mathbf{x}) = \frac{x_j}{\sqrt{\frac{1}{N}\sum_k x_k^2 + \varepsilon}}\gamma_j \tag{17}$$

where $\varepsilon, \gamma_j, \beta_j \in \mathbb{R}$. Affine transformations such as the multiplicative weighting of the output or the subtraction of the mean value are linear operations that can be attributed by the ε-LRP rule. Normalization, on the other hand, is non-linear and requires separate considerations. As such, we focus on the following function:

$$f_j(\mathbf{x}) = \frac{x_j}{g(\mathbf{x})} \tag{18}$$

where $g(\mathbf{x}) = \sqrt{\text{Var}[\mathbf{x}] + \varepsilon}$ or $g(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_k x_k^2 + \varepsilon}$.

The work (Ali et al., 2022) demonstrates that when linearizing LayerNorm at $\mathbf{x}$, the bias term absorbs most of the relevance equal to $\text{Var}[\mathbf{x}]/(\text{Var}[\mathbf{x}] + \varepsilon)$, effectively absorbing 99% of the relevance with commonly used values of $\varepsilon = 10^{-6}$ and $\text{Var}[\mathbf{x}] = 1$. Hence, a linearization at $\mathbf{x}$ is not meaningful. As a solution, (Ali et al., 2022) proposes to regard $g(\mathbf{x})$ as a constant, which transforms the normalization operation (18) into a (linear) element-wise operation, on which the identity rule (9) can be applied, as discussed in Appendix A.2.2. In the following, we prove that this heuristic can be derived from the Deep Taylor Decomposition framework.

Proposition 3.4 *Decomposing LayerNorm or RMSNorm by a Taylor decomposition (4) with reference point $\boldsymbol{0}$ (without bias or distributing the bias uniformly) yields the identity relevance propagation rule:*

$$R_i^{l-1} = R_i^l \tag{19}$$

There is no bias that absorbs relevance. The proof is given in Appendix A.3.4. This rule enforces a strict notion of conservation, while being highly efficient by excluding normalization operations from the computational graph. Experiments in Section 4.1 provide evidence that this simplification is faithful.

3.4. Understanding Latent Features

As we iterate through each layer during the attribution process with AttnLRP, we obtain relevance values for each latent neuron as a by-product. Ranking this latent relevance enables us to identify neurons and layers that are most influential for the reasoning process of the model (Achtibat et al., 2023). The subsequent step is to reveal the concept that is represented by each neuron by finding the most representative reference samples that explain the neuron's encoding. A common technique is Activation Maximization (ActMax) (Nguyen et al., 2016), where input samples are sought that give rise to the highest activation value. We follow up on these observations and present the following strategy for understanding latent features: (1) Collect prompts that lead to the highest activation of a unit. (2) Explain the unit's activation using AttnLRP, allowing to narrow down the relevant input tokens for the chosen unit.

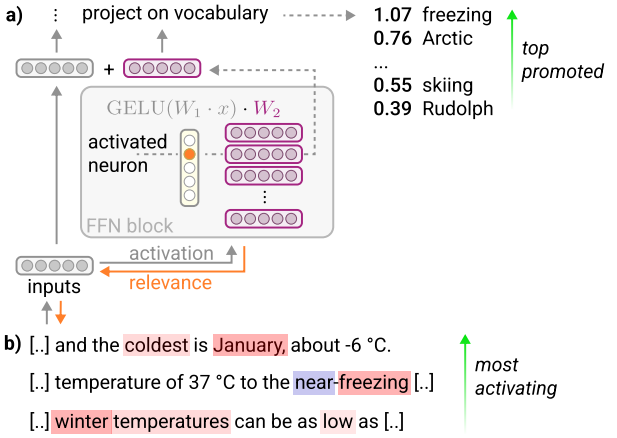

Figure 3. There are two approaches for understanding knowledge neurons: (a) Neuron `3948` at the last non-linearity in FFN 17 of the Phi-1.5 model selects a weight row to add to the residual stream. This weight row projected on the vocabulary spans topics about ice, cold places and winter sport. (b) Sentences that maximally activate this neuron contain references about coldness. Attributing the neuron with AttnLRP highlights the most relevant tokens inside the input sentences. Inspired by (Voita et al., 2023).

In this work, we concentrate on knowledge neurons (Dai et al., 2022; Voita et al., 2023) that are situated at the last non-linearity in FFN layers $\mathbf{z} = \text{GELU}(\mathbf{W}_1\mathbf{x})$. These neurons possess intriguing properties, as shown in Figure 3: They encode factual knowledge and upon activation, the corresponding row of the second weight matrix $\mathbf{W}_2$ is added to the residual stream directly influencing the output distribution of the model. By projecting this weight row onto the vocabulary, a distribution of the most probable tokens across the vocabulary is obtained (Geva et al., 2022). Applying AttnLRP on ActMax reference samples and projecting the weight row on the vocabulary allow us to understand in which context a neuron activates and how its activation influences the prediction of the next token. In contrast to (Ali et al., 2022), AttnLRP also allows analyzing the key and value linear layers inside attention modules.

4. Experiments

Our experiments aim to answer the following questions:

(Q1) How faithful are our explanations compared to other state-of-the-art approaches?

(Q2) How efficient is LRP compared to perturbation-based methods?

(Q3) Can we understand latent representations and interact with LLMs?

4.1. Evaluating Explanations (Q1)

A reliable measure of faithfulness of an explanation are input perturbation experiments (Samek et al., 2017; Hedström et al., 2023). This approach iteratively substitutes the most important tokens in the input domain with a baseline value. If the attribution method accurately identified the most important tokens, the model's confidence in the predicted output should rapidly decrease. The other way around, perturbing the least relevant tokens first, should not affect the model's prediction and result in a slow decline of the model's confidence. For more details, see Appendix B.2. Despite its drawbacks, such as potentially introducing out-of-distribution manipulations (Chang et al., 2018) and sensitivity towards the chosen baseline value, this approach is widely adopted in the community. (Brocki and Chung, 2023; Blücher et al., 2024) have addressed this criticism and introduced an enhanced metric by quantifying the area between the least and most relevant order perturbation curves to obtain a robust measure. Hence, we will employ this improved metric to measure faithfulness. Appendix Figure B.5 illustrates a typical perturbation curve.

In order to assess plausibility, we utilize the SQuAD v2 Question-Answering (QA) dataset (Rajpurkar et al., 2018), which includes a ground truth mask indicating the correct answer within the question. We calculate attributions for accurately answered questions and determine the top-1 accuracy of the most relevant token and the Intersection over Union (IoU) between the positive attribution values and the ground truth mask. This approach assumes that the model solely relies on the information provided in the ground truth mask, which is not entirely accurate but sufficient for identifying a trend.

4.1.1. Baselines

We evaluate the faithfulness on two self-attention models, a ViT-B-16 (Dosovitskiy et al., 2021) on ImageNet (Deng et al., 2009) classification and the LLaMa 2-7b (Touvron et al., 2023) model on IMDB movie review (Maas et al., 2011) classification as well as next word prediction of Wikipedia (Wikimedia Foundation, 2023). Additional results for ViT-L-16 and ViT-L-32 are in Appendix Table B.6. To assess plausability, we employ two instruction-finetuned models on the SQuAD v2 dataset: the MoE model Mixtral 8x7b (Jiang et al., 2024) and the encoder-decoder model Flan T5-XL (Chung et al., 2022). We denote our method as AttnLRP and compare it against a broad spectrum of methods including Input×Gradient (I×G), Integrated Gradients (IG), SmoothGrad (SmoothG), Attention Rollout (AttnRoll), Gradient-weighted Attention Rollout (G×AttnRoll) and Conservative Propagation (CP)-LRP. As explained in Appendix A.2.3, we propose to apply the γ-rule for AttnLRP in the case of ViTs. For better compari-

Table 1. Faithfulness scores as area between the least and most relevant order perturbation curves (Blücher et al., 2024) on different models and datasets. To assess plausibility, the (top-1) accuracy along with the IoU in parentheses are depicted for SQuAD v2. Methods marked with $(*)$ have been proposed here. Additional results for ViT-L-16 and ViT-L-32 are in Appendix Table B.6.

Methods	**ViT-B-16**	**LLaMa 2-7b**		**Mixtral 8x7b**	**Flan-T5-XL**
	ImageNet ↑	IMDB ↑	Wikipedia ↑	SQuAD v2 ↑	SQuAD v2 ↑
Random	0.01 ± 0.01	-0.01 ± 0.05	-0.07 ± 0.13	0.03 (0.09)	0.03 (0.08)
Input×Grad (Simonyan et al., 2014)	0.80 ± 0.03	0.12 ± 0.05	0.18 ± 0.13	0.56 (0.35)	0.60 (0.39)
IG (Sundararajan et al., 2017)	1.54 ± 0.03	1.23 ± 0.05	4.05 ± 0.13	0.68 (0.44)	0.10 (0.16)
SmoothGrad (Smilkov et al., 2017)	-0.04 ± 0.03	0.25 ± 0.05	-2.22 ± 0.14	0.47 (0.24)	0.05 (0.09)
GradCAM (Chefer et al., 2021b)	0.27 ± 0.04	-0.82 ± 0.05	2.01 ± 0.15	0.82 (0.72)	0.81 (0.70)
AttnRoll (Abnar and Zuidema, 2020)	1.31 ± 0.03	-0.64 ± 0.05	-3.49 ± 0.15	0.05 (0.10)	0.02 (0.08)
Grad×AttnRoll (Chefer et al., 2021a)	2.60 ± 0.03	1.61 ± 0.05	9.79 ± 0.14	0.91 (0.40)	**0.94** (0.53)
AtMan (Deb et al., 2023)	0.70 ± 0.02	-0.20 ± 0.05	3.31 ± 0.15	0.86 (**0.83**)	0.88 (0.80)
KernelSHAP (Lundberg and Lee, 2017)	4.71 ± 0.03	-	-	-	-
CP-LRP (ε-rule, Ali et al. (2022))	2.53 ± 0.02	1.72 ± 0.04	7.85 ± 0.12	0.50 (0.40)	0.91 (0.83)
CP-LRP (γ-rule for ViT, as proposed here)*	6.06 ± 0.02	-	-	-	-
AttnLRP (ours)*	$\mathbf{6.19} \pm 0.02$	$\mathbf{2.50} \pm 0.05$	$\mathbf{10.93} \pm 0.13$	**0.96** (0.72)	**0.94** (**0.84**)

son, we also included an enhanced CP-LRP baseline, which also uses the γ-rule in the ViTs experiment. The LRP variants introduced by (Voita et al., 2021; Chefer et al., 2021b) are excluded due to numerical instabilities observed in preliminary experiments, see also Appendix A.2.1. Further, we utilize the Grad-CAM adaptation described in (Chefer et al., 2021b). Specifically, we weight the last attention map with the gradient. For a fair comparison, we attribute all methods without the softmax at the classification output, except AtMan which relies on it. KernelSHAP is only evaluated on vision transformers due to prohibitive computational costs on larger LLMs. Finally, we expand upon AtMan by incorporating it into encoder-decoder models by suppressing tokens in all self-attention layers within the encoder, while only doing so in cross-attention layers within the decoder. For AtMan, SmoothGrad and Rollout-methods we perform a hyperparameter sweep over a subset of the dataset. More details about baseline methods and the hyperparameter search are in Appendix A.1 and B.3. We illustrate example heatmaps for SQuAD v2 in Appendix B.7.

4.1.2. Discussion

In Table 1, we can observe that AttnLRP consistently outperforms all the state-of-the-art methods in terms of faithfulness. In models with a higher number of non-linearities (higher complexity), AttnLRP demonstrates substantially higher accuracy compared to CP-LRP. While the relative improvement to CP-LRP is 3% for Flan-T5-XL, which only utilizes standard attention layers, AttnLRP achieves a remarkable 46% improvement over CP-LRP in terms of top-1 accuracy in Mixtral 8x7b, that incorporates additional expert layers with softmax non-linearities and FFN layers with non-linear weighting. In Appendix B.4, we discuss the architectural differences and conduct an ablation study on different model components to demonstrate this effect. We also observe that gradient-based approaches significantly suffer from noisy attributions, as reflected by the low faithfulness and illustrated in example heatmaps in Appendix B.7. CP-LRP with ε applied on all layers (as proposed in Ali et al. (2022)), also suffers from noisy gradients in ViTs. Applying instead the γ-rule for CP-LRP and AttnLRP in ViTs improves the faithfulness substantially. Whereas AtMan and GradCAM do not perform well in unstructured tasks, *i.e.*, next word prediction or classification, they achieve a high score in QA tasks. While G×AttnRoll better reflects the model behavior compared to AtMan and GradCAM, it is affected by considerable background noise, resulting in a low IoU score in the SQuAD v2 dataset.

4.2. Computational Complexity and Memory Consumption (Q2)

Table 2 illustrates the computational complexity and memory consumption of a single LRP-based attribution and linear-time perturbation, such as AtMan or a Shapley-based method (Fatima et al., 2008). Linear-time perturbation requires N_T forward passes, but has only a memory requirement of $\mathcal{O}(1)$. Since LRP is a backpropagation-based method, gradient checkpointing (Chen et al., 2016) techniques can be applied. In checkpointing, LRP requires two forward and one backward pass, while the memory requirement scales logarithmic with the number of layers. In Appendix B.8, we benchmark energy, time and memory consumption of LRP against perturbation-based methods

Table 2. Computational and memory complexity of LRP-based and linear-time perturbation methods measured w.r.t. a single forward pass. N_L: number of layers, N_T: number of tokens

Methods	Computational Complexity	Memory Consumption
LRP Checkpointing	$\mathcal{O}(1)$	$\mathcal{O}(\sqrt{N_L})$
Perturbation (linear)	$\mathcal{O}(N_T)$	$\mathcal{O}(1)$

across context- and model-sizes.

4.3. Understanding & Manipulating Neurons (Q3)

In our investigation, we use the Phi-1.5 model (Li et al., 2023), which has a transformer-based architecture with a next-word prediction objective. We obtain reference samples for each knowledge neuron by collecting the most activating sentences over the Wikipedia summary dataset (Scheepers, 2017).

To illustrate, we consider the prompt: '`The ice bear lives in the`' which gives the corresponding prediction: '`Arctic`'. Using AttnLRP, we determine the most relevant layers for predicting '`Arctic`' as well as the specific neurons within the FFN layers contributing to this prediction. Our analysis reveals that the most relevant neurons after the first three layers are predominantly situated within the middle layers. Notably, one standout neuron `#3948` in `layer 17` activates on reference samples about cold temperatures, as depicted in Figure 3. This observation is further validated by projecting the weight matrix of the second FFN layer onto the vocabulary. The neuron shifts the output distribution of the model to cold places, winter sports and animals living in cold regions.

Analogously, for the prompt '`Children love to eat sugar and`' with the prediction '`sweets`', the most relevant neuron's (`layer 18`, neuron `#5687`) projection onto the vocabulary signifies a shift in the model's focus towards the concept of candy, temptation and sweetness in the vocabulary space. We interact with the model by deactivating neuron `#3948`, and strongly amplifying the activation of neuron `#5687` in the forward pass. This manipulation yields the following prediction change:

Prompt: `Ice bears live in the`
Prediction: `sweet, sugary treats of the candy store.`

We further notice that neuron `#4104` in `layer 17` encodes for dryness, thirst and sand. Increasing its activation changes the output to '`desert`' (illustrated in Figure 2).

With AttnLRP, we are able to trace the most important neurons in models with billions of parameters. This allows us to systematically navigate the latent space to enable targeted modifications to reduce the impact of certain concepts (for example, '`coldness`') and enhance the presence of other concepts (for example, '`dryness`'), resulting in discernible output changes. Such an approach holds significant implications for transformer-based models, which have been difficult to manipulate and explain due to inherent opacity and size.

5. Conclusion

We have extended the Layer-wise Relevance Propagation framework to non-linear attention, proposing novel rules for the softmax and matrix-multiplication step and providing interpretations in terms of Deep Taylor Decomposition. Our AttnLRP method stands out due to its unique combination of simplicity, faithfulness, and efficiency. We demonstrate its applicability both for LLMs as well as ViTs, utilizing the denoising effect of the γ-rule. In contrast to other backpropagation-based approaches, AttnLRP enables the accurate attribution of neurons in latent space (also within the attention module), thereby introducing novel possibilities for real-time model interaction and interpretation.

Limitations & Open Problems

Adjusting the γ-parameter in ViTs remains crucial to achieve accurate attributions. To reduce memory consumption, the impact of quantization on attributions and custom GPU kernels for LRP rules should be investigated.

Acknowledgements

We extend our heartfelt gratitude to Leila Arras for her invaluable feedback and to Johanna Vielhaben for improving our faithfulness metric. We also thank Patrick Kahardipraja, Daniel Becking and Maximilian Ernst for their insightful comments.

Impact Statement

This work establishes the foundations that make it possible to systematically analyze and debug transformer-based AI systems, thereby minimizing the occurrence of false or misleading outputs (hallucination) and mitigating biases that may arise from training data or algorithmic processes. Particularly, it opens up the door for future applications of transformer-based AI systems in critical domains such as healthcare and finance, where the ability to explain the model behavior is often a (legal) requirement. The high computational efficiency of our method significantly reduces the energy usage and consequently also the financial overhead and environmental impact associated with the explanation, which will result in a broader adoption of XAI for transformers.

References

Abnar, S. and Zuidema, W. H. (2020). Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.

Achtibat, R., Dreyer, M., Eisenbraun, I., Bosse, S., Wiegand, T., Samek, W., and Lapuschkin, S. (2023). From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019.

Ali, A., Schnake, T., Eberle, O., Montavon, G., Müller, K.-R., and Wolf, L. (2022). Xai for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning*, pages 435–451. PMLR.

Anders, C. J., Neumann, D., Samek, W., Müller, K.-R., and Lapuschkin, S. (2021). Software for dataset-wide xai: from local explanations to global insights with zennit, corelay, and virelay. *arXiv preprint arXiv:2106.13200*.

Arras, L., Osman, A., and Samek, W. (2022). Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. (2017). The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pages 342–350. PMLR.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W. (2016). Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pages 63–71. Springer.

Blücher, S., Vielhaben, J., and Strodthoff, N. (2024). Decoupling pixel flipping and occlusion strategy for consistent xai benchmarks. *arXiv preprint arXiv:2401.06654*.

Brocki, L. and Chung, N. C. (2023). Feature perturbation augmentation for reliable evaluation of importance estimators in neural networks. *Pattern Recognition Letters*, 176:131–139.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660.

Chang, C.-H., Creager, E., Goldenberg, A., and Duvenaud, D. (2018). Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*.

Chefer, H., Gur, S., and Wolf, L. (2021a). Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406.

Chefer, H., Gur, S., and Wolf, L. (2021b). Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791.

Chen, T., Xu, B., Zhang, C., and Guestrin, C. (2016). Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. (2022). Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Deb, M., Deiseroth, B., Weinbach, S., Schramowski, P., and Kersting, K. (2023). Atman: Understanding transformer predictions through memory efficient attention manipulation. *arXiv preprint arXiv:2301.08110*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Ding, Y., Liu, Y., Luan, H., and Sun, M. (2017). Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159.

Dombrowski, A.-K., Anders, C. J., Müller, K.-R., and Kessel, P. (2022). Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR*.

Fatima, S. S., Wooldridge, M., and Jennings, N. R. (2008). A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673–1699.

Fedus, W., Dean, J., and Zoph, B. (2022). A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*.

Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457.

Fryer, D., Strümke, I., and Nguyen, H. (2021). Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9:144352–144360.

Geva, M., Caciularu, A., Wang, K., and Goldberg, Y. (2022). Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

Geva, M., Schuster, R., Berant, J., and Levy, O. (2021). Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Gildenblat, J. (2020. Accessed on Dec 01, 2023). Exploring explainability for vision transformers. https://jacobgil.github.io/deeplearning/vision-transformer-explainability.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., and Höhne, M. M. M. (2023). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al. (2020). Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., and Lee, Y. T. (2023). Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Lundberg, S. M. and Lee, S. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (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.

Mao, C., Jiang, L., Dehghani, M., Vondrick, C., Sukthankar, R., and Essa, I. (2021). Discrete representations strengthen vision transformer robustness. In *International Conference on Learning Representations*.

Miglani, V., Yang, A., Markosyan, A., Garcia-Olano, D., and Kokhlikyan, N. (2023). Using captum to explain generative language models. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 165–173.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29.

Pahde, F., Yolcu, G. Ü., Binder, A., Samek, W., and Lapuschkin, S. (2023). Optimizing explanations by network canonization and hyperparameter search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3818–3827.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.

Scheepers, T. (2017). Improving the compositionality of word embeddings. Master's thesis, Universiteit van Amsterdam, Science Park 904, Amsterdam, Netherlands.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Shaham, U., Ivgi, M., Efrat, A., Berant, J., and Levy, O. (2023). Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Voita, E., Ferrando, J., and Nalmpantis, C. (2023). Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*.

Voita, E., Sennrich, R., and Titov, I. (2021). Analyzing the source and target contributions to predictions in neural machine translation. In *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 1126–1140.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. 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 11–20.

Wikimedia Foundation (2023. Accessed on Dec 01, 2023). Wikimedia downloads. https://dumps.wikimedia.org.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference Computer Vision - ECCV 2014*, pages 818–833.

Zhang, B. and Sennrich, R. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Appendix

A. Appendix I: Methodological Details

This appendix provides further details on the methods presented in the paper. In particular, we focus on the AttnLRP method, provide implementation details, discuss the stability of the bias term, highlight the difference between other LRP variants, discuss the noise problem in Vision Transformers and illustrate the effects of temperature scaling on attributing the softmax function. Finally, we provide proofs for the four propositions presented in the main paper.

A.1. Details on Baseline Methods

In the following, we present an overview of the baseline methods and their hyperparamter choices.

A.1.1. Input × Gradient

Gradients are one of the most straightforward approaches to depict how sensitive the trained model is with respect to each individual given feature (traditionally of the input space). By weighting the gradient with the input features, the model is locally linearized (Simonyan et al., 2014):

$$\mathrm{I} \times \mathrm{G}(\mathbf{x}) = \frac{\partial f_c(\mathbf{x})}{\partial \mathbf{x}} \times \mathbf{x} \tag{20}$$

Due to the gradient shattering effect (Balduzzi et al., 2017) which is a known phenomenon (especially in the ReLU-based CNNs), heatmaps generated by I×G are very noisy, making them in many cases not meaningful.

A.1.2. Integrated Gradients

To tackle the noisiness of I×G, the idea to integrate gradients along a trajectory has been proposed. Here, the gradients of different (m) interpolated versions of the input $\mathbf{x}$, noted by $\mathbf{x}'$, are integrated as (Sundararajan et al., 2017):

$$\begin{aligned}\mathrm{IG}(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}') \int_{\alpha=0}^{1} \frac{\partial f_j(\mathbf{x}' + \alpha \times (\mathbf{x}-\mathbf{x}'))}{\partial \mathbf{x}} d\mathbf{x} \\ &\approx (\mathbf{x}-\mathbf{x}') \sum_{k=1}^{m} \frac{\partial f_j(\mathbf{x}' + \frac{k}{m} \times (\mathbf{x}-\mathbf{x}'))}{\partial \mathbf{x}} \times \frac{1}{m}\end{aligned} \tag{21}$$

We utilize `zennit` (Anders et al., 2021) and its default settings to compute Integrated Gradients attribution maps *i.e.* $m = 20$.

A.1.3. SmoothGrad

A different technique towards the reduction of noisy gradients is smoothing the gradients (Smilkov et al., 2017) through generating (m) various samples in the neighborhood of input $\mathbf{x}$ as $\mathbf{x}_\varepsilon = \mathbf{x} + \mathcal{N}(\mu, \sigma^2)$ and computing the

average of all gradients:

$$\text{SmoothGrad}(\mathbf{x}) = \frac{1}{m}\sum_{1}^{m}\frac{\partial f_j(\mathbf{x}_\varepsilon)}{\partial \mathbf{x}_\varepsilon} \tag{22}$$

In this work, we set $\mu = 0$ and perform a hyperparameter search for σ to find the optimal parameter. We utilize `zennit` (Anders et al., 2021) and its default settings to compute SmoothGrad attribution maps *i.e.* $m = 20$.

A.1.4. Attention Rollout

Self-Attention rollout (Abnar and Zuidema, 2020) capitalizes on the intrinsic nature of the attention weights matrix $\mathbf{A} \in \mathbb{R}^{b\times s_q \times s_k}$ as a representative measure of token importance. It generates a $s_q \times s_k$ matrix where each row is normalized to form a probability distribution, representing the importance of each query token to all key tokens. The attention scores along the head dimension are averaged:

$$\bar{\mathbf{A}} = \mathbb{E}_b[\mathbf{A}]$$

where $\mathbb{E}_b$ denotes the expectation along the head dimension b of the attention map. To compute the relevance of hidden layer tokens (h) to the original input tokens (i), an iterative multiplication of the attention matrices on the left side is sufficient. Hence, the key dimension represents the inputs and the query dimension the outputs. To account for the residual connection through which the information of the previous tokens flows, an identity matrix $\mathbf{I}$ is added:

$$\mathbf{R}_k^{h,i} = (\mathbf{I} + \bar{\mathbf{A}}^{h,h}) \cdot \mathbf{R}_{k-1}^{h,i} \tag{23}$$

where $k = 1$ corresponds to the input layer and $\mathbf{R}_0^{h,i}$ is initialized with the identity matrix $\mathbf{I}$, h denotes the hidden feature space, and i stands for input dimension.

(Chefer et al., 2021a) build upon self-attention rollout and weights the attention matrix with the gradient. Additionally, the weighted attention map is denoised by computing the mean value of only positive values.

$$\bar{\mathbf{A}} = \mathbb{E}_b[(\nabla\mathbf{A} \odot \mathbf{A})^+]$$

For encoder-decoder models, (Chefer et al., 2021a) present several additional considerations that are not mentioned here for brevity.

(Gildenblat, 2023) notes, that the rollout attributions can further be improved by discarding outlier values. For that, we define a discard threshold $dt \in [0, 1]$ used to compute the quantile $Q(dt)$, where dt represents the proportion of the data below the quantile *e.g.* with cumulative distribution function $P(\bar{\mathbf{A}} \leq Q(dt)) = dt$.

$$\bar{\mathbf{A}}_{m,n} = \begin{cases} 0 & \text{if } \bar{\mathbf{A}}_{m,n} > Q(dt) \\ \bar{\mathbf{A}}_{m,n} & \text{otherwise} \end{cases}$$

A.1.5. AtMan

AtMan (Deb et al., 2023) perturbs the pre-softmax activations along the k-dimension:

$$\mathbf{H} = \mathbf{Q} \cdot \mathbf{K}^\top$$
$$\tilde{\mathbf{H}} = \mathbf{H} \odot (\mathbf{1} - \mathbf{p}^i)$$

where $\mathbf{H} \in \mathbb{R}^{b\times s_q \times s_k}$, and $\mathbf{1} \in [1]^{b\times s_q \times s_k}$ a matrix containing only 1. $\mathbf{p}^i$ denotes a matrix $\in \mathbb{R}^{b\times s_q \times s_k}$ with

$$\mathbf{p}^i_{lmn} = \begin{cases} p \text{ for } n = i \\ 0 \text{ for } n \neq i \end{cases}$$

Thus, for a single token $i \in \{1, 2, ..., N\}$, we suppress all values along the column/key-dimension with a suppression factor p. The suppression factor is a hyperparameter that must be tuned to the dataset and model. For ViTs, additional cosine similarities are computed to suppress correlated tokens as detailed in (Deb et al., 2023). For that, an additional hyperparameter denoted as t for threshold must be optimized in ViTs only.

A.1.6. KernelSHAP

LIME computes attributions by fitting an additive surrogate model (Ribeiro et al., 2016). KernelSHAP (Lundberg and Lee, 2017) is a special case of LIME, that sets the loss function, weighting kernel and regularization terms of LIME such that LIME recovers Shapley values. Hence, KernelSHAP allows theoretically to obtain Shapley Values more efficiently than directly computing Shapley Values.

To apply KernelSHAP in the vision domain, we divide the input image into N super-pixels using the Simple Linear Iterative Clustering (SLIC) algorithm (Achanta et al., 2012). We use `captum` (Kokhlikyan et al., 2020) with its default settings to compute the attributions *i.e.* number of samples per attributions set to 2000 and baseline value set to 0.5. A baseline value of 0 resulted in lower faithfulness. For SLIC, we set $N = 100$ with compactness set to 10.

A.2. Details on AttnLRP

This section provides more details on AttnLRP and justifies the specific parameter choices made in our work (*e.g.*, use of γ-LRP in Vision Transformers).

A.2.1. Conservation & Numerical Stability of Bias Terms

The total relevance R_j of a layer output, *i.e.* linearized function $f_j(\mathbf{x}) = \sum_i \mathbf{J}_{ji}\ x_i + \tilde{b}_j$, is computed by summing the contributions of the input variables $R_{i\leftarrow j}$, represented by $\mathbf{J}_{ji}\ x_i$, and adding the contribution of the bias

term $R_{b\leftarrow j}$, represented by $\tilde{b}_j$.

$$R_j = \sum_i R_{i\leftarrow j} + R_{b\leftarrow j},$$

The relevance of the input variables is solely determined by the input variables themselves

$$R_i = \sum_j R_{i\leftarrow j} = \sum_j \mathbf{J}_{ji}\, x_i \frac{R_j}{f_j(\mathbf{x})},$$

while the relevance of the bias term itself is calculated as

$$R_{b\leftarrow j} = \tilde{b}_j \frac{R_j}{f_j(\mathbf{x})}.$$

If we want to compute the relevance of the input variables R_i while ensuring strict adherence to the conservation property (3), we must exclude the bias term, so that it does not absorb part of the relevance. In the literature, we find two common practices: Either distributing the bias term uniformely on the input variables (Binder et al., 2016; Voita et al., 2021) or applying the identity rule (Ding et al., 2017; Chefer et al., 2021b). Both approaches can lead to severe numerical instabilities in specific cases that are challenging to identify. Therefore, we will dedicate some time to explain the issue in greater detail.

Remark A.2.1 *Enforcing strict conservation* (3) *on a function, where*

$$\exists i, j \in \mathbb{N} : x_i = 0 \wedge f_j(\boldsymbol{x}) \neq 0,$$

by distributing the relevance of the bias term of its linearization uniformly on the input variables or applying the identity rule with $i = j$ may lead to numerical instabilities.

Distributing the bias term: We can distribute the relevance value of the bias term uniformly across the input variables by assuming that the bias term is part of the input variables:

$$R_j^l = \sum_i \tilde{R}_{i\leftarrow j}^{(l-1,l)} = \sum_i^N \left(R_{i\leftarrow j}^{(l-1,l)} + \frac{R_{b\leftarrow j}^{(l-1,l)}}{N} \right),$$

where N represents the number of input variables. Hence,

$$R_i^l = \sum_j \tilde{R}_{i\leftarrow j}^{(l-1,l)} = \sum_j \left(\mathbf{J}_{ji}\, x_i + \frac{\tilde{b_j}}{N} \right) \frac{R_j^l}{f_j(\mathbf{x})}.$$

However, we may encounter numerical instabilities, but these effects will only become visible in the next sequential relevance propagation at the prior layer, not at this layer yet. For example in the softmax function, we may encounter a situation where $\exists i, j \in \mathbb{N}$ with $x_i^{l-1} = 0$ but $f_j^l(\mathbf{x}) > 0$. If a non-zero relevance value from layer l is assigned to $f_j^l(\mathbf{x})$, then its relevance value is propagated to the input variable x_i^{l-1} through the relevance message:

$$\tilde{R}_{i\leftarrow j}^{(l-1,l)} = \frac{\tilde{b}_j}{N} \frac{R_j^l}{f_j^l(\mathbf{x}) + \varepsilon}$$

Assuming we apply the ε-rule in succession, the relevance in the prior layer is given by:

$$R_{k\leftarrow i}^{(l-2,l-1)} = \mathbf{J}_{ik}\, x_k^{l-2} \frac{R_i^{l-1}}{0 + \varepsilon}$$

Here, we divide by $x_i^{l-1} = f_i^{l-1} = 0$. Since ε is very small, this term explodes and causes numerical instabilities. Hence, assigning a non-zero relevance value to an input that equals zero leads to numerical instabilities. Note, that these instabilities would not occur if $R_i^{l-1} = 0$ *e.g.* $R_{i\leftarrow j}^{(l-1,l)} = 0$. Functions where $f(0) = 0$ do not encounter this issue, because zero output activations will not receive any relevance in following layers *e.g.* $R_j^l = 0$ using all rules described in this paper.

Applying the identity rule: Alternatively, we can apply the identity rule as follows:

$$R_i^{l-1} = R_i^l$$

Here, numerical instabilities might also arise in the subsequent relevance propagation, not at this layer. With the same reasoning as before, the identity rule propagates a non-zero relevance value to an input variable that is zero.

Omitting the bias term: For the sake of completeness, we mention that omitting the bias term entirely is also an option. In this case, the relevance propagation equation is:

$$R_i^{l-1} = \sum_j \mathbf{J}_{ji}\, x_i \frac{R_j^l}{\sum_i \mathbf{J}_{ji}\, x_i + \varepsilon}$$

Here, we no longer divide by the original function $f_j(\mathbf{x})$, but by its linearization without the bias term. However, it is important to ensure that no sign flips occur, as $\sum_i \mathbf{J}_{ji}\, x_i$ might have a different sign than $f_j(\mathbf{x})$.

Remark A.2.2 *Enforcing strict conservation* (3) *by omitting the bias term of a linearization* (4) *can lead to sign flips in the relevance scores.*

Summary: In summary, applying the identity rule, distributing its relevance value uniformly across the input variables or omitting the bias term completely are possible approaches, but they have their considerations and potential challenges. Regarding the softmax non-linearity, (Voita et al., 2021) distributes the bias term equally on all input variables, while (Ding et al., 2017; Chefer et al., 2021b) apply the element-wise identity rule (9). Both variants can lead to severe numerical instabilities.

A.2.2. Highlighting the difference between various LRP methods

In Table A.3, we illustrate the different strategies employed for LRP in the past.

Softmax: (Voita et al., 2021) linearizes at $\mathbf{x}$ but distributes the bias term equally on all input variables, while (Ding et al., 2017; Chefer et al., 2021b) apply the element-wise identity rule (9). More specifically, (Ding et al., 2017) did not discuss the softmax function explicitly, but they skip all non-linear activation functions. Therefore, we assume that (Ding et al., 2017) applies the identity rule also to the softmax function. Both variants enforce a strict notion of the conservation principle (3), but can lead to severe numerical instabilities as discussed in Appendix A.2.1. (Ali et al., 2022) regards the attention matrix $\mathbf{A}$ in Equation (11) as constant, attributing relevance solely through the value path by stopping the relevance flow through the softmax. Consequently, the query and key matrices can no longer be attributed, which reduces the faithfulness and makes latent explanations in query and key matrices infeasible. Finally, AttnLRP linearizes at $\mathbf{x}$ with a bias term that absorbs part of the relevance. The presence of a bias term in AttnLRP is justified because the softmax function yields a value of 1/N even when the input is zero. This is analogous to a bias term and is necessary to account for this behavior. This ensures not only numerically robust attributions, but also improves the faithfulness considerably.

In Figure A.4, we illustrate different attribution maps for all four options to handle the softmax function. The given section is from the Wikipedia article on Mount Everest. The model is expected to provide an answer for the question `'How high did they climb in 1922?'` and for the correctly predicted next token `3` of the answer `'According to the text, the 1922 expedition reached 8,'` is the attribution computed by initializing the relevance at the predicted token with its logit value.

While the relevance values for AttnLRP or CP-LRP are between $[-4, 4]$, distributing the bias uniformely on the input variables or applying the identity rule leads to an explosion of the relevances between $[-10^{15}, 10^{15}]$. As a consequence, the heatmaps resemble random noise. AttnLRP highlights the correct token the strongest, while CP-LRP focuses strongly on the start-of-sequence `<s>` token and exhibits more background noise *e.g.* irrelevant tokens such as 'Context', 'attracts', 'Everest' are highlighted, while AttnLRP does not highlight them or assigns negative relevance. In Appendix B.7, we compare also other baseline methods. Note, that the model attends to numerous tokens within the text which enables it to derive conclusions. Consequently, an attribution that reflects the model behavior will highlight more than just the single accurate answer token. The faithfulness experiments in Table 1 demonstrate, that AttnLRP captures the model reasoning most accurately.

Matrix Multiplication: Applying the ε-rule (8) on *bi-linear* matrix multiplication (11) violates the conservation property (3) as proved in Appendix A.3.5. To the best of our knowledge, (Ding et al., 2017) applies the standard ε-rule. (Voita et al., 2021) utilizes the z^+-rule (25), that similar to the ε-rule also violates the conservation property (3) in bi-linear matrix multiplication (proof in Appendix A.3.5 is valid for z^+-rule). While (Chefer et al., 2021b) also applies the ε-rule, an additional normalization step is performed by dividing both arguments by the summation of its absolute values. This ensures conservation but is not conform with the DTD framework. (Ding et al., 2017; Chefer et al., 2021b) set the ε parameter to 0, which may increases numerical instabilities. Hence, we call their LRP variants in Table A.3 0-LRP.

Since (Ali et al., 2022) regards the softmax output as constant and does not propagate relevance through it, the matrix multiplication is not bi-linear anymore, but becomes linear. Hence, the application of the ε-rule does not violate the conservation principle and attributes only the value path. (Ali et al., 2022) sets the ε parameter to zero, hence we call their LRP variant in Table A.3 0-LRP.

Finally, AttnLRP also applies the ε-rule on bi-linear matrix multiplication. In addition, a novel uniform rule (14) derived from the DTD framework is incorporated that ensures conservation and high faithfulness.

Layer Normalization: The works (Chefer et al., 2021b; Ali et al., 2022) and AttnLRP apply the identity rule on normalization functions (18), while using the ε-rule on all linear components of LayerNorm, if applicable. More specifically, (Ali et al., 2022) proposes to regard $g(\mathbf{x})$ in (18) as a constant, which transforms the normalization operation, and hence the complete LayerNorm layer, into a linear layer, on which the ε-rule is applied. However, this is similar to applying the identity rule (9) on the normalization itself, because it becomes element-wise with a single input and output variable (see Section 3.2.2), while applying the ε-rule on all other linear components of LayerNorm. (Voita et al., 2021) linearizes at $\mathbf{x}$ and distributes the bias term equally on all input variables, which can lead to numerical instabilities as discussed in Appendix A.2.1.

Vision Transformer: The studies by (Ding et al., 2017; Voita et al., 2021; Ali et al., 2022) concentrate on the attribution in natural language processing (NLP) models and do not address vision transformers. Their methodologies, as demonstrated in Table 1 (and Appendix A.2.1), fail when implementing the ε-rule, leading to gradient shattering and low faithfulness. To mitigate noisy attributions,

(Chefer et al., 2021b) suggests employing on-top of LRP an attention rollout (Abnar and Zuidema, 2020) procedure which is additionally enhanced via gradient weighting. This yields an approximation of the mean squared relevance value, which diverges from the originally defined notion of "relevance" or "importance" of additive explanatory models. Subsequent empirical observations by (Chefer et al., 2021a) revealed that an omission of LRP-inspired relevances and a sole reliance on a positive mean-weighting of the attention's activation with the gradient improved the faithfulness. Though, this approach can only attribute positively, does not consider counteracting evidence, and does not allow to attribute latent neurons outside the attention's softmax output. AttnLRP, in contrast, adopts the γ-rule instead of the ε-rule in linear layers, achieving highly faithful attributions without the necessity for a rollout mechanism. However, the γ parameter must be tuned to the model and dataset to obtain optimal attributions, as discussed in Appendix A.2.3.

A.2.3. Tackling Noise in Vision Transformers

Since backpropagation-based attributions utilize the gradient, they may produce noisy attributions in models with many layers, where gradient shattering and noisy gradients appear (Balduzzi et al., 2017; Dombrowski et al., 2022). Hence, various adaptions of the ε-LRP rule were developed to strengthen the signal-to-noise ratio by dampening counter-acting activations (Bach et al., 2015; Montavon et al., 2019). Here, we use the generalized γ-rule that encompasses all other proposed rules in the literature (Montavon et al., 2019). Let z_{ij} be the contribution of input i to output j, *e.g.* $\mathbf{W}_{ji}x_i$, and z_j the neuron output activation. Then depending on the sign of z_j:

$$R_{i\leftarrow j}^{(l,\,l+1)} = \begin{cases} \frac{z_{ij}+\gamma z_{ij}^{+}}{z_j+\gamma\sum_k z_{kj}^{+}} R_j^{l+1} & \text{if } z_j > 0 \\ \frac{z_{ij}+\gamma z_{ij}^{-}}{z_j+\gamma\sum_k z_{kj}^{-}} R_j^{l+1} & \text{else} \end{cases} \tag{24}$$

with $\gamma \in \mathbb{R}^{>0}$, $(\cdot)^+ = \max(\cdot\,, 0)$ and $(\cdot)^- = \min(\cdot\,, 0)$. If $\gamma = \infty$, it is equivalent to the LRP z^+-rule, which is given as

$$R_{i\leftarrow j}^{(l,\,l+1)} = \frac{(w_{ij}x_i)^+}{z_j^+} R_j^{l+1} \tag{25}$$

by only taking into account positive contributions $z_j^+ = \sum_i (w_{ij}x_i)^+$ with $(\cdot)^+ = \max(0, \cdot)$.

Remarkably, our observations reveal that attributions in LLMs demonstrate high sparsity and lack visible noise, while ViTs are susceptible to gradient shattering. We hypothesize that the discrete nature of the text domain may affect robustness (Mao et al., 2021). Therefore, we only apply the γ-rule in ViTs in the convolutional and linear FFN layers outside the attention module. To further increase the faithfulness, the γ-rule can be also applied on softmax layers. Since the output of the softmax is always greater than zero, we apply the simplified z^+-rule (special case of γ-rule). The z^+-rule applied on a linearization (4) for softmax results in:

$$R_i^{l-1} = \sum_j (\mathbf{J}_{ji}\, x_i)^+ \frac{R_j^l}{\sum_k (\mathbf{J}_{jk}\, x_k)^+ + \tilde{b}_j^+} \tag{26}$$

This formula is computationally more expensive to evaluate than the original rule for softmax derived in Proposition 3.1. Should efficiency be a priority, it is recommended to bypass the softmax layer as proposed in CP-LRP, which prevents relevance from passing through the softmax function and reduces gradient shattering caused by this layer. Then, for all other components of the model, AttnLRP rules are recommended, with the application of the γ-rule to linear layers only. This is especially true, given that the discrepancy in faithfulness between AttnLRP and γCP-LRP is minimal for standard vision architectures (Dosovitskiy et al., 2021) evaluated in Table 1 and Table B.6, that incorporate only standard attention and FFN layers.

A.2.4. Impact of Temperature Scaling on the Softmax Rule

Temperature scaling controls the entropy within the softmax probability distribution, thereby influencing the predictability of subsequent next token predictions at the classification output. A high temperature value tends to flatten the softmax output distribution (more randomness), whereas a small temperature parameter sharpens the distribution (less randomness). This scaling is done by dividing the input $\mathbf{x}$ by the temperature $T \in \mathbb{R}$ prior to applying the softmax function.

$$s_j(\mathbf{x}) = \frac{e^{x_j/T}}{\sum_i e^{x_i/T}}$$

Recall that the derivative of the softmax function has two cases, which depend on the output j and input i indices:

$$\frac{\partial s_j}{\partial x_i} = \begin{cases} s_j(1-s_j) & \text{for } i = j \\ -s_j s_i & \text{for } i \neq j \end{cases}$$

In the scenario where $s_j \approx 1$, the derivative for $i = j$ vanishes. This occurs *e.g.* for extremely low temperature values or exceptionally high confidence in the model's classification output. This poses an issue for the Deep Taylor Decomposition (Montavon et al., 2017) derived in Section 3.1.1, because DTD decomposes the softmax function by utilizing the gradient (jacobian) term $\mathbf{J}_{ji}x_i$ for calculating attributions. If the gradient vanishes, the bias term will

Table A.3. Conceptual differences between various-LRP methods and their implications. "Layer Normalization" refers here only to the normalization function (18) itself and not to the learnable parameters of LayerNorm or RMSNorm.

Methods	Softmax	Matrix Multiplication	Layer Normalization
(Ding et al., 2017)	Identity rule ⇒ unstable (Appendix A.2.1)	0-LRP (bi-linear) ⇒ violates conservation	not available
(Voita et al., 2021)	Taylor decomposition at **x** (distributes the bias uniformly) ⇒ unstable (Appendix A.2.1)	z^+-LRP (bi-linear) ⇒ violates conservation	Taylor decomposition at **x** (distributes the bias uniformly) ⇒ unstable (Appendix A.2.1)
(Chefer et al., 2021b)	Identity rule ⇒ unstable (Appendix A.2.1)	0-LRP & post-hoc normalization (bi-linear) ⇒ ensures conservation	Identity rule ⇒ ensures conservation & faithful
(Ali et al., 2022)	Regarded as constant ⇒ stable & no attribution inside attention module	0-LRP (linear only) ⇒ ensures conservation	Identity rule ⇒ ensures conservation & faithful
AttnLRP	Taylor decomposition at **x** (with bias) ⇒ stable & faithful	ε-LRP & uniform rule (bi-linear) ⇒ ensures conservation & faithful	Identity rule ⇒ ensures conservation & faithful

capture all the relevance, stopping the relevance flow altogether. This effect is also generally described by (Shrikumar et al., 2017).

Within the attention mechanism, this limitation is circumvented by multiplying the softmax output with the value path, ensuring that relevance is transmitted via the uniform rule to the value path, akin to CP-LRP (refer to Appendix A.2.2). However, in instances where the softmax function is utilized independently, this becomes problematic as the relevance flow could be distorted.

To see this, consider Proposition 3.1 (13): R_i^{l-1} might be zero if $R_j^l = 0 \; \forall j \neq i$ with $s_i = 1$:

$$R_i^{l-1} = x_i(R_i^l - s_i \sum_j R_j^l) = x_i(R_i^l - R_i^l) = 0$$

Therefore, we suggest utilizing an increased temperature scaling value when explaining the softmax classification output to prevent that the softmax saturates *i.e.* the gradient vanishes. Nonetheless, the attribution of the classification output has not been investigated in this work (the softmax layer is always removed and LRP only applied to the logit outputs). An analysis of these effects remain an interesting topic for future work.

A.3. Proofs

In the following, we provide proofs for the rules presented in the main paper, and that the application of the ε-rule on bi-linear matrix multiplication violates the conservation property.

A.3.1. Proposition 3.1: Decomposing Softmax

In this subsection, we demonstrate the decomposition of the softmax function by linearizing (4) it at **x** . We begin by considering the softmax function:

$$s_j(\mathbf{x}) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

The derivative of the softmax has two cases, which depend on the output and input indices i and j:

$$\frac{\partial s_j}{\partial x_i} = \begin{cases} s_j(1 - s_j) & \text{for } i = j \\ -s_j s_i & \text{for } i \neq j \end{cases}$$

Consequently, a Taylor decomposition (4) yields:

$$f_j(\mathbf{x}) = s_j \left(x_j - \sum_i s_i x_i \right) + \tilde{b}_j$$

We differentiate between two cases, namely (i) when we attribute relevance from output j to input $i \neq j$ and (ii) when we attribute from output j to input $i = j$.

$$R_{i\leftarrow j}^{(l-1,l)} = \begin{cases} (x_i - s_i x_i)\, R_i^l & \text{for } i = j \\ -s_i x_i\, R_j^l & \text{for } i \neq j \end{cases}$$

Applying equation (5), we obtain:

$$R_i^{l-1} = \sum_j R_{i\leftarrow j}^{(l-1,l)} = x_i(R_i^l - s_i \sum_j R_j^l)$$

In Appendix A.2.4, we discuss the implications of vanishing gradients and temperature scaling on attributing the softmax function.

A.3.2. Proposition 3.2: Decomposing Multiplication

The aim in is subsection is to decompose the multiplication of N input variables.

$$f_j(\mathbf{x}) = \prod_i^N x_{ji}$$

We start by performing a Taylor decomposition (4), then we derive the same decomposition with Shapley.

Taylor decomposition: The derivative is

$$\frac{\partial f_j}{\partial x_{ji}} = \prod_{k\neq i}^N x_{jk}$$

Consequently, a Taylor decomposition (4) at $\mathbf{x}$ yields

$$f_j(\mathbf{x}) = \sum_i^N \frac{\partial f_j}{\partial x_{ji}} x_{ji} + \tilde{b}_j = N \prod_k^N x_{jk} + \tilde{b}_j = N f_j(\mathbf{x}) + \tilde{b}_j$$

We can either omit the bias term or equally distribute it on the input variables to strictly enforce the conservation property (3). Here, we demonstrate how to distribute the bias term uniformly.

$$R_{ji\leftarrow j}^{(l-1,l)} = \left(f_j + \frac{\tilde{b}_j}{N}\right) \frac{R_j^l}{N f(\mathbf{x})_j + \tilde{b}_j} = \frac{1}{N} R_j^l$$

Since each input with index ji at layer $l-1$ is only connected to one output with index j at layer l, we have only a single relevance propagation message. Hence, it follows from Equation (2):

$$R_{ji}^{l-1} = R_{ji\leftarrow j}^{(l-1,l)} = \frac{1}{N} R_j^l$$

For omitting the bias term, repeat the proof with $\tilde{b}_j = 0$.

Shapley: The Shapley value (Lundberg and Lee, 2017) is defined as:

$$\phi_i(f) = \sum_{\substack{S\subseteq N \\ i\notin S}} \frac{|S|!(N-|S|-1)!}{N!} \left(f(S\cup\{i\}) - f(S)\right) \tag{27}$$

where $\phi_i(v)$ is the Shapley value of the feature i and value function f. N denotes the set of all features, and S denotes a feature subset (coalition).

With respect to multiplication, zero is the absorbing element. Hence, we choose zero as our baseline value, and the Shapley value function becomes:

$$f(S\cup\{i\}) = \prod_k x_k$$
$$f(S) = 0$$
$$f(S\cup\{i\}) - f(S)) = \prod_k x_k$$

The symmetry theorem (Fryer et al., 2021) of Shapley states that the contributions of two feature values i and l should be the same if they contribute equally to all possible coalitions

$$f(S\cup\{i\}) = f(S\cup\{l\})$$
$$\forall S \subseteq \{1,2,...N\}\backslash\{i,l\}$$

then $\phi_i(f) = \phi_l(f)$. In addition, the efficiency theorem (Fryer et al., 2021) states that the output contribution is distributed equally amongst all features. Hence, the output contribution is equal to the sum of coalition values of all features i,

$$\sum_i \phi_i(f) = f(N)$$

Both theorems are applicable and hence it follows:

$$\phi_i(f) = \frac{1}{N} f(N)$$

In the case of LRP, we identify $f(N)$ as R_j^l and $\phi_i(f)$ as R_{ji}^{l-1}.

A.3.3. Proposition 3.3: Decomposing bi-linear Matrix Multiplication

Consider the equation for matrix multiplication, where we treat the terms as single input variables by substituting them with $\mathbf{u}_{jip} = \mathbf{A}_{ji}\mathbf{V}_{ip}$

$$\mathbf{O}_{jp} = \sum_i \mathbf{A}_{ji}\mathbf{V}_{ip} = \sum_i \mathbf{u}_{jip}$$

In this case, the function already is in the form of an additive decomposition (1). Therefore,

$$R_{jip\leftarrow jp}^{(l-1,l)} \propto \mathbf{u}_{jip}$$

This can also be seen, by noticing that $\sum_i \mathbf{u}_{jip}$ is a linear operation, since a single variable is left. Hence, it can be regarded as a linear layer (6) characterized with constant weights of one and a bias of zero. We already derived the solution to applying a linearization (4) to a linear layer: the ε-rule (8). Therefore, the solution is:

$$R^{(l-1,l)}_{jip\leftarrow jp} = \mathbf{u}_{jip}\frac{R^l_{jp}}{\mathbf{O}_{jp}+\varepsilon} = \mathbf{A}_{ji}\mathbf{V}_{ip}\frac{R^l_{jp}}{\mathbf{O}_{jp}+\varepsilon}$$

Since each input with index jip at layer $l-1$ is only connected to one output with index jp at layer l, we have only a single relevance propagation message. Hence, it follows from equation (2):

$$R^{l-1}_{jip} = R^{(l-1,l)}_{jip\leftarrow jp}$$

Next, we decompose the individual terms $\mathbf{u}_{jip}$ using the uniform rule from the previous Section A.3.2 to obtain relevance messages for $\mathbf{A}_{ji}$:

$$R^{(l-1,l-1)}_{ji\leftarrow jip} = \frac{1}{2}R^{l-1}_{jip}$$

Each input $\mathbf{A}_{ji}$ is connected to p outputs $\mathbf{u}_{jip}$. Hence, to obtain the relevance values attributed to $\mathbf{A}_{ji}$, we must aggregate all relevance messages from output jip to inputs ji via Equation (2):

$$R^{l-1}_{ji} = \sum_p R^{(l-1,l-1)}_{ji\leftarrow jip} = \sum_p \frac{1}{2}R^{l-1}_{jip}$$

$$R^{l-1}_{ji} = \sum_p \mathbf{A}_{ji}\mathbf{V}_{ip}\frac{R^l_{jp}}{2(\mathbf{O}_{jp}+\varepsilon)}$$

Because $\varepsilon \ll |\mathbf{O}_{jp}|$, we simplify the final solution:

$$R^{l-1}_{ji} = \sum_p \mathbf{A}_{ji}\mathbf{V}_{ip}\frac{R^l_{jp}}{2\,\mathbf{O}_{jp}+\varepsilon}$$

The proof for $\mathbf{V}_{ip}$ follows a similar approach by summing over the j indices instead of p. In Appendix A.3.5, we proof that this rule does not violate the conservation property (3) in contrast to the standard ε-rule.

A.3.4. Proposition 3.4: Layer Normalization

Consider layer normalization of the form

$$f_j(\mathbf{x}) = \frac{x_j}{g(\mathbf{x})}$$

where $g(\mathbf{x}) = \sqrt{\mathrm{Var}[\mathbf{x}]+\varepsilon}$ or $g(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_k x_k^2+\varepsilon}$. The derivative is

$$\frac{\partial f_j}{\partial x_i} = \frac{1}{g(\mathbf{x})^2}\begin{cases} g(\mathbf{x}) - x_j\frac{\partial g(\mathbf{x})}{\partial x_i} & \text{for } i=j \\ -x_j\frac{\partial g(\mathbf{x})}{\partial x_i} & \text{for } i\neq j \end{cases} \tag{28}$$

In LayerNorm (Ba et al., 2016), we assume for simplicity $\mathbb{E}[\mathbf{x}]=0$, then the partial derivative simplifies to

$$\mathbb{V}[\mathbf{x}] = \mathbb{E}[\mathbf{x}^2] - \mathbb{E}[\mathbf{x}]^2 = \mathbb{E}[\mathbf{x}^2]$$
$$\frac{\partial \mathbb{V}[\mathbf{x}]}{\partial x_i} = \frac{2}{N}x_i$$

Further, the partial derivative of RMSNorm (Zhang and Sennrich, 2019) is

$$\frac{\partial \text{RMSNorm}}{\partial x_i} = \frac{x_i}{\sqrt{N\sum_k x_k}}$$

At reference point $\tilde{x}_i = 0$, the diagonal elements in Equation (28) $i \neq j$ are zero, yielding the Taylor decomposition:

$$f_i(\mathbf{x}) = \frac{\partial f_i}{\partial x_i}\Big|_{\tilde{x}_i=0} x_i + \tilde{b}_i = \frac{x_i}{\varepsilon} + \tilde{b}_i$$

To enforce a strict notion of the conservation property (3), the bias term $\tilde{b}_i$ can be excluded or evenly distributed across the input variables. Because we have only a single input variable, the bias can be considered as part of x_i.

$$R^{l-1}_i = \left(\frac{x_i}{\varepsilon}+\tilde{b}_i\right)\frac{R^l_i}{\frac{x_i}{\varepsilon}+\tilde{b}_i} \tag{29}$$

Since there is only one input variable and one output, the decomposition is equivalent to the identity function, as discussed in the Section 3.2.2 about component-wise non-linearities. Thus, we conclude that the identity rule applies in this case.

$$R^{l-1}_i = R^l_i \tag{30}$$

Note, that this rule is numerically stable because $f_j(0)=0$ as discussed in Section A.2.1.

A.3.5. Violation of the Conservation Property in Bi-Linear Matrix Multiplication

In the following we proof that the application of the ε-rule (8) without the uniform rule (14) on bi-linear matrix multiplication violates the conservation property (3). We reiterate and generalize the Lemma 3 of (Chefer et al., 2021b) which establishes that 0-LRP ($\varepsilon=0$) violates conservation.

Recall, that matrix multiplication is defined as:

$$\mathbf{O}_{jp} = \sum_i \mathbf{A}_{ji}\mathbf{V}_{ip}$$

The ε-rule is the solution to applying a linearization (4) to a linear layer (6). For computing relevance values for $\mathbf{A}_{ji}$ using the ε-rule, we treat $\mathbf{V}_{ip}$ as a constant weight matrix

with zero bias, and similarly for attributing $\mathbf{A}_{ji}$. The derived relevance propagation rules are given by:

$$\tilde{R}^{l-1}_{ji}(\mathbf{A}_{ji}) = \sum_p \mathbf{A}_{ji}\mathbf{V}_{ip}\frac{R^l_{jp}}{\mathbf{O}_{jp}+\varepsilon}$$
$$\tilde{R}^{l-1}_{ip}(\mathbf{V}_{ip}) = \sum_j \mathbf{A}_{ji}\mathbf{V}_{ip}\frac{R^l_{jp}}{\mathbf{O}_{jp}+\varepsilon}$$

The conservation property (3) states, that the total relevance at layer l must be equal to the total relevance at layer $l-1$.

The total relevance at layer l is given by

$$R^l = \sum_{j,p} R^l_{jp}$$

and the total relevance at layer $l-1$ is computed by:

$$R^{l-1} = \sum_{j,i}\tilde{R}^{l-1}_{ji} + \sum_{i,p}\tilde{R}^{l-1}_{ip} = 2\sum_{j,p}\frac{\mathbf{O}_{jp}}{\mathbf{O}_{jp}+\varepsilon}R^l_{jp} \approx 2R^l$$

with $\varepsilon \ll |\mathbf{O}_{jp}|$. This results in a violation of the conservation property as $R^l \neq R^{l-1}$. However, by employing Proposition 3.3 (15), that is a sequential application of the ε-rule and uniform rule (14), we ensure conservation by dividing with the factor 2:

$$R^{l-1} = \sum_{j,i} R^{l-1}_{ji} + \sum_{i,p} R^{l-1}_{ip} = \sum_{j,p}\frac{\mathbf{O}_{jp}}{\mathbf{O}_{jp}+\varepsilon}R^l_{jp} \approx R^l$$

It is evident that ε absorbs a negligible proportion of the relevance to safeguard numerical stability. The proof is also valid for the z^+-rule (25), where only positive contributions are taken into consideration.

B. Appendix II: Experimental Details

In the following sections, we provide additional details about the experiments performed.

B.1. Models and Datasets

For ImageNet faithfulness, we utilized the pretrained Vision Transformer B-16, L-16 and L-32 weights of the PyTorch model zoo (Paszke et al., 2019). We randomly selected 3200 samples (fixed set for all baselines) such that the standard error of mean converges to below 1% of the mean value.

For Wikipedia and IMDB faithfulness, we evaluated the pretrained LLaMa 2-7b hosted on huggingface (Wolf et al., 2019) on 4000 randomly selected validation dataset samples (fixed set for all baselines). For SQuAD v2, we utilize the pretrained Flan-T5 and Mixtral 8x7b weights hosted on huggingface. Further, for Wikipedia next word prediction we evaluated the model on a context size of 512 (from beginning of article until context length is reached), while the context size in SQuAD v2 varies between 169 to 4060. Although Flan-T5 was trained on a smaller context size of 2000 tokens, the relative positional encoding allows it to handle longer context sizes with at least 8192 tokens (Shaham et al., 2023).

All computations are performed in the Brain Floating Point (bfloat16) half-precision format to save memory consumption. bfloat16 trades precision for a higher dynamic range than standard float16, and hence prevents numerical errors due to overflow. In this regard, the impact of quantized number formats on (Attn)LRP attributions remains a topic to be investigated. In addition, all linear weights in Mixtral 8x7b are quantized to the 4 bit integer format using *bitsandbytes* (Dettmers et al., 2024) (but computation still performed in bfloat16).

SQuAD v2 encompasses numerous questions that are either unanswerable or subject to incorrect predictions by the model. Consequently, only instances where the model accurately predicts the correct response are considered. Additionally, only the testset is utilized to mitigate a potential overfitting bias during the training phase, if applicable. Finally, for SQuAD v2 top-1 accuracy and IoU, we utilized the following prompt:

```
Context:  [text of dataset sample]
Question:  [question of dataset sample]
Answer:
```

Flan-T5 does not require a system prompt, while for Mixtral 8x7b we use before the context the system prompt:

```
Use the context to answer the question.
Use few words.
```

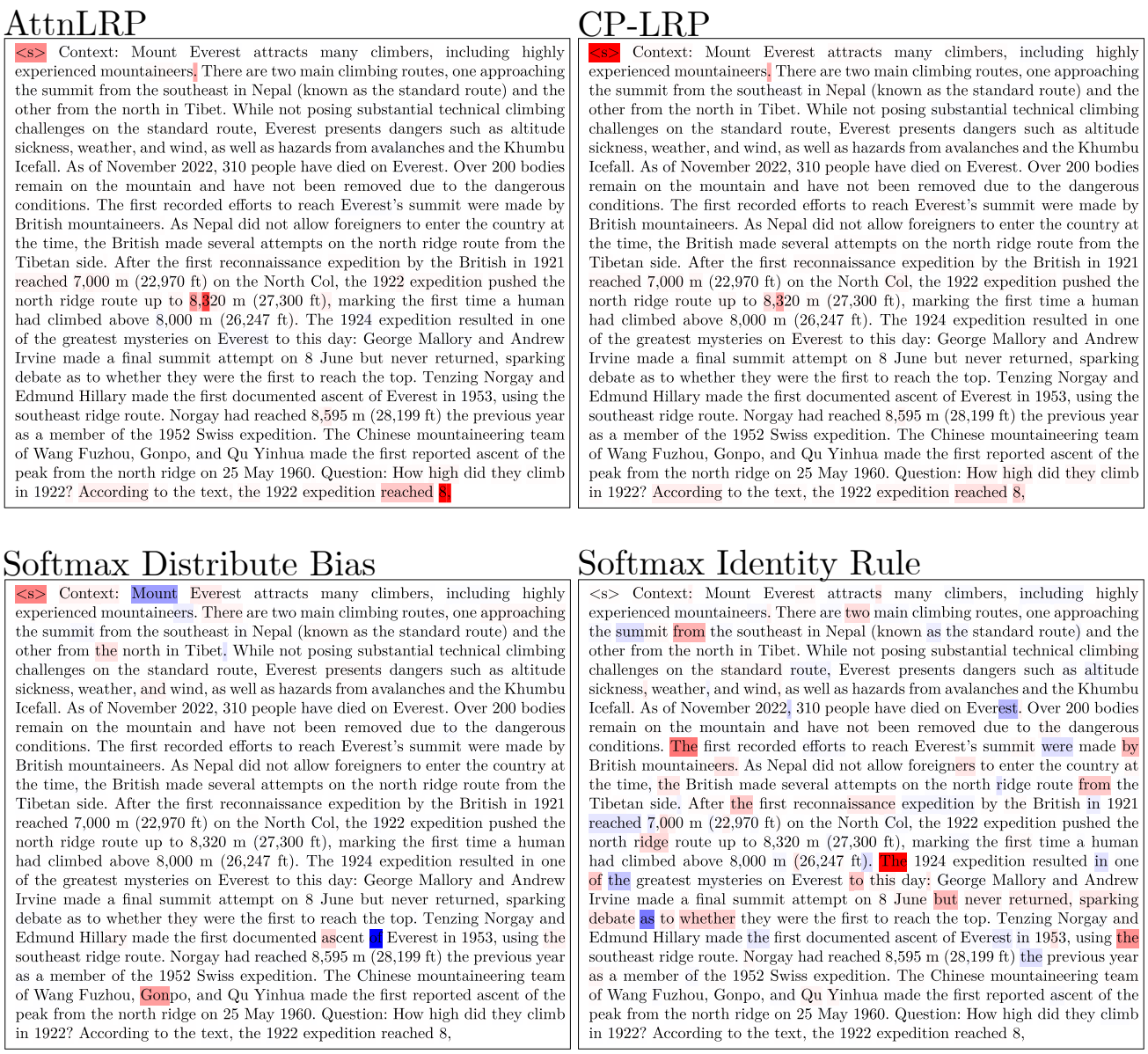

Figure A.4. Comparison of four different LRP variants computed on a LLaMa 2-7b model. The given section is from the Wikipedia article on Mount Everest. The model is expected to provide the next answer token for the question `'How high did they climb in 1922? According to the text, the 1922 expedition reached 8,'`. For the correctly predicted token `3` the attribution is computed. Distributing the bias uniformely on the input variables (Softmax Distribute Bias) or applying the identity rule (Softmax Identity Rule) leads to numerical instabilities. For "Softmax Distribute Bias" and "Softmax Identity Rule", we applied AttnLRP rules on all layers except for the softmax function. AttnLRP highlights the correct token the strongest, while CP-LRP focuses strongly on the start-of-sequence `<s>` token and exhibits more background noise *e.g.* irrelevant tokens such as 'Context', 'attracts', 'Everest' are highlighted, while AttnLRP does not highlight them or assigns negative relevance.

Because Flan-T5 typically provides the correct answer directly, we explain the first token of the answer only. Conversely, Mixtral 8x7b generates full sentences; within these, we identify the positions of the answer tokens and explain all tokens that constitute the correct answer only. To achieve this, we calculate heatmaps for each answer token and add these heatmaps to produce the final heatmap. For gradient-based methods, this process can be parallelized by initiating the backward pass at the designated token positions with the logit output for LRP and with the value 1 for all other baselines, while initializing the remaining output tokens with zero.

For IMDB, we added a last linear layer to a frozen

LLaMa 2-7b model and finetuned only the last layer, which achieves 93% accuracy on the validation dataset.

If we encountered NaN values for a sample, we removed it from the evaluation. This happened for Grad $\times$ AttnRollout and AtMan in the Wikipedia dataset. However, the standard error of the mean remains small, as can be seen in Table 1.

B.2. Input Perturbation Metrics

In the following, we summarize the perturbation process introduced by (Samek et al., 2017) in a condensed manner.

Given an attributions map $R_i^l(x_i)$ per input features $\mathbf{x} = \{x_i\}_{i=1}^N$ in layer l. $\mathcal{H}$ denotes a set of relevance values for all input features x_i:

$$\mathcal{H} = (R_0^0(x_0), R_1^0(x_1), ..., R_{N-1}^0(x_{N-1})) \tag{31}$$

Then, the *flipping* perturbation process iteratively substitutes input features with a baseline value $\mathbf{b} \in \mathbb{R}^N$ (the baseline might be zero, noise generated from a Gaussian distribution, or pixels of a black image in the vision task). Another reverse equivalent variant, referred as *insertion*, begins with a baseline $\mathbf{b}$ and reconstructs the input $\mathbf{x}$ stepwise. The function performing the perturbation is denoted by $\mathbf{g}^F$ for flipping and $\mathbf{g}^I$ for insertion. The perturbation procedure is either conducted in a MoRF (Most Relevant First) or LeRF (Least Relevant First) manner based on the sorted members of $\mathcal{H}$. Regardless of the replacement function, the MoRF and LeRF perturbation processes can be defined as recursive formulas at step $k = \{0, 1, ..., N-1\}$:

$$\text{MoRF Pert. Process} = \begin{cases} \mathbf{x}_{MoRF}^0 = \mathbf{x} \\ \mathbf{x}_{MoRF}^k = \mathbf{g}^{(F|I)}(\mathbf{x}_{MoRF}^{k-1}, \mathbf{b}) \\ \mathbf{x}_{MoRF}^{N-1} = \mathbf{b} \end{cases}$$

where $\mathbf{x}_{MoRF}^k$ denotes the perturbed input feature $\mathbf{x}$ at step k in MoRF process.

$$\text{LeRF Pert. Process} = \begin{cases} \mathbf{x}_{LeRF}^0 = \mathbf{b} \\ \mathbf{x}_{LeRF}^k = \mathbf{g}^{(F|I)}(\mathbf{x}_{LeRF}^{k-1}, \mathbf{b}) \\ \mathbf{x}_{LeRF}^{N-1} = \mathbf{x} \end{cases}$$

where $\mathbf{x}_{LeRF}^k$ denotes the perturbed input feature $\mathbf{x}$ at step k in LeRF process.

Results of these processes are perturbed input sets of $\mathcal{X}_{MoRF}^F = (\mathbf{x}_{MoRF}^0, \mathbf{x}_{MoRF}^1, ..., \mathbf{x}_{MoRF}^{N-1})$ and $\mathcal{X}_{LeRF}^F = (\mathbf{x}_{LeRF}^0, \mathbf{x}_{LeRF}^1, ..., \mathbf{x}_{LeRF}^{N-1})$. By feeding these sets to the model and computing the corresponding logit output f_j, a curve will be induced and consequently the area A under the curve can be calculated:

$$A_{MoRF}^F = A_{LeRF}^I = \frac{1}{N} \sum_{k=0}^{N-1} f_j(\mathbf{x}_{MoRF}^k) \tag{32}$$

where $\mathbf{x}_{MoRF}^k \in \mathcal{X}_{MoRF}^F$ or $\mathbf{x}_{MoRF}^k \in \mathcal{X}_{LeRF}^I$.

It is notable that the area below the least relevant order insertion curves are identical to the most relevant order flipping curves and that the area below the least relevant order flipping curves are identical to the most relevant order insertion curves. Hence, by using $\mathcal{X}_{MoRF}^I$, $A_{MoRF}^I = A_{LeRF}^F$ can be computed similarly. A faithful explainer results in a low value of A_{MoRF}^F or A_{LeRF}^I. Further, a faithful explainer is expected to have large A_{LeRF}^F or A_{MoRF}^I values.

Ultimately to reduce introducing out-of-distribution manipulations and the sensitivity towards the chosen baseline value, the work of (Blücher et al., 2024) proposes to leverage both insights and to obtain a robust measure as

$$\Delta A^F = A_{LeRF}^F - A_{MoRF}^F$$
$$\Delta A^I = A_{MoRF}^I - A_{LeRF}^I$$

where a higher score signifies a more faithful explainer.

We performed all faithfulness perturbations with a baseline value of zero. In the case of LLMs, we aggreated the relevance for each token and flipped the entire embedding vector of input tokens to the baseline value. For ViTs, we used the relevances of the input pixels and flipped input pixels to the baseline value.

B.3. Hyperparameter search for Baselines

As describes in Appendix A.1, several baseline attribution methods have hyperparameters that must be tuned to the datasets. The default parameters are described in Appendix A.1, and to reduce the search space, we optimize a subset of the hyperparameters. The hyperparameters of SmoothGrad ($\sigma \in [0.01, 0.25]$), AtMan (suppression value $\in [0.1, 1.0]$ and threshold $\in [0, 1.0]$), AttnRoll (discard threshold $\in [0.90, 1.00]$), and G$\times$AttnRoll (discard threshold $\in [0.90, 1.00]$) are selected to be optimized. The used hyperparameters for the perturbation experiments are available in the captions of Tables B.6, B.7, B.8 and B.9.

For LLMs, we have not noticed a significant impact on the heatmaps for different discard threshold values of G$\times$AttnRoll. For AttnRoll, the impact is minimal. Hence, we choose the default value of 1 (nothing is discarded, as proposed in the original works (Abnar and Zuidema, 2020; Chefer et al., 2021a)).

Regarding SQuAD v2, we set AtMan's $p = 0.7$ for Mixtral

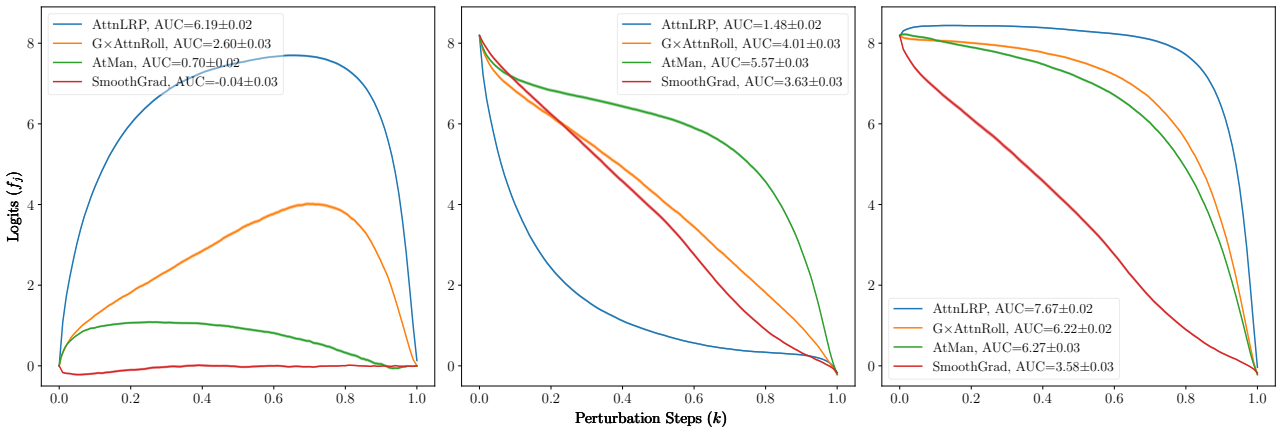

Figure B.5. Comparison of the AttnLRP (ours) with the γ-rule, Grad×AttnRoll (Chefer et al., 2021a), AtMan (Deb et al., 2023), and SmoothGrad (Smilkov et al., 2017) techniques through the perturbation experiment (faithfulness) on the ViT-B-16 using 3200 random samples of ImageNet. From left to right, the plots correspond to $f_j(\mathcal{X}^F_{LeRF}) - f_j(\mathcal{X}^F_{MoRF})$ (large area is good), $f_j(\mathcal{X}^F_{MoRF})$ (steep decline is good), and $f_j(\mathcal{X}^F_{LeRF})$ (slow decline is good). "AUC" denotes the Area under Curve.

8x7b and $p = 0.9$ for Flan-T5-XL. For SmoothGrad, we set $\sigma = 0.1$ for Mixtral 8x7b and Flan-T5-XL.

B.4. Impact of Model Architectural Choices on AttnLRP Performance

We evaluated AttnLRP on three model classes that incorporate different types of layers.

Flan-T5: This encoder-decoder architecture employs self-attention and cross-attention layers (33). The FFN layers are a sequential application of linear layers with GELU non-linearities inbetween.

LLaMa 2: This decoder architecture utilizes only self-attention layers (33). However, in the FFN layers, we have an additional element-wise non-linear weighting with a SiLU non-linearity (34).

Mixtral 8x7b: This mixture of experts model uses self-attention layers (33) and FFN layers with non-linear weighting (34) like the LLaMa 2. In addition, there are FFN routing layers with a softmax weighting (35).

$$\text{Attention: } \text{Softmax}(\mathbf{W}_q\,\mathbf{x}\,(\mathbf{W}_k\,\mathbf{x})^\top)\,\mathbf{W}_v\,\mathbf{x} \tag{33}$$

$$\text{FFN} \times \text{Non-Linearity: } \text{SiLU}(\mathbf{W}_1\,\mathbf{x}) \odot \mathbf{W}_2\,\mathbf{x} \tag{34}$$

$$\text{Routing: } \sum_i \text{Softmax}(\text{TopK}(\mathbf{W}_g\,\mathbf{x}))_i\,\text{FFN}_i(\mathbf{x}) \tag{35}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_g$ are linear weight parameters, $(\odot)$ is element-wise multiplication, $i \in \mathbb{N}$ the number of expert FFN layers and TopK is returning the top-k elements.

In Table B.4, we study the impact of AttnLRP rules w.r.t. CP-LRP on all three different layer types (33), (34) and (35). We start as baseline with all rules of CP-LRP on all layer types, then we successively substitute CP-LRP rules with AttnLRP rules for specific layer types.

For CP-LRP, we use the rules described in Appendix A.2.2. In the original work of (Ali et al., 2022), FFN layers weighted with non-linearities and routing layers are not discussed. Analogously to the argumentation in their work, we regard the non-linearity as constant weight and attribute only through the FFN path using the ε-rule.

As demonstrated in Table B.4, the application of AttnLRP rules enhances the performance across all layers, regardless of their type. Moreover, the rate of improvement increases with the number of non-linearities present in the model.

B.5. LRP Composites for ViT

Applying the ε-rule on all linear layers inside LLMs is sufficient to obtain faithful and noise-free attributions. However, for the vision transformers, we apply the γ-rule on all linear layers (including the convolutional layers) outside the attention module. Since the γ-rule has a hyperparameter, the work (Pahde et al., 2023) proposed to tune the parameter using a grid-search. This optimization search (or in an LRP context known as composite search) is computational highly demanding.

The vision transformer consists of many linear layers. Our proposed approach is to use different γ values across different layer types.

Table B.4. This Table is an extension of Table 1: Faithfulness scores as area between the least and most relevant order perturbation curves on LLaMa 2 alongside the top-1 accuracy and IoU in parenthesis for Flan-T5 and Mixtral 8x7b. We start as baseline with all rules of CP-LRP on all layer types, then we successively substitute CP-LRP rules with AttnLRP rules for specific layer types if they exist in the model. We observe that AttnLRP's improvement is not confined to the attention mechnism alone, but all layers that contain operations that are not attributable with other LRP variants. The more complex the architecture, the better the performance of AttnLRP compared to CP-LRP.

Method on Layer	**LLaMa 2-7b**		**Mixtral 8x7b**	**Flan-T5-XL**
	IMDB ↑	Wikipedia ↑	SQuAD v2 ↑	SQuAD v2 ↑
Baseline (All Layers)				
CP-LRP	1.72	7.85	0.50 (0.40)	0.90 (0.83)
+ Attention Mechanism				
AttnLRP	2.09	9.49	0.70 (0.53)	0.94 (0.84)
+ FFN $\times$ Non-Linearity				
AttnLRP	2.50	10.93	0.78 (0.57)	-
+ Routing Layer				
AttnLRP	-	-	0.96 (0.72)	-

According to (Vaswani et al., 2017) the attention module consists of several linear layers which we refer to as *LinearInputProjection*.

$$\mathbf{Q} = \mathbf{W}_q\mathbf{X} + \mathbf{b}_q$$

$$\mathbf{K} = \mathbf{W}_k\mathbf{X} + \mathbf{b}_k$$

$$\mathbf{V} = \mathbf{W}_v\mathbf{X} + \mathbf{b}_v$$

In the attention layer, after the softmax (11), there exists another linear layer performing the output projection back into the residual stream, denoted as *LinearOutputProjection*:

$$\mathbf{y} = \mathbf{W}_o\mathbf{O} + \mathbf{b}_o$$

The other layers in the whole network, will be referred to as *Linear*.

The perturbation experiment had been conducted over these layers using different types of rules including Epsilon, ZPlus, Gamma, and AlphaBeta (with $\alpha = 2$ and $\beta = 1$ according to (Montavon et al., 2019)).

The most faithful composite, that we obtained for AttnLRP and CP-LRP, is in Table B.5. More details over the statistics of the conducted experiments are available in Figures B.14, B.15, B.16, B.17, B.18.

B.6. Additional Perturbation Evaluations on Vision Transformers

Table B.6 presents additional perturbation results for the vision transformers ViT-L-16 and ViT-L-32 evaluated on ImageNet. Our method surpasses all comparative baselines, while the proposed enhancement, γCP-LRP (which applies the γ-rule across all linear layers for CP-LRP (Ali et al., 2022)), remains highly competitive. In more complex model architectures that incorporate a greater variety of non-linearities, our method demonstrates more superiority, as elaborated in Appendix B.4. Tuning the γ-parameter for γCP-LRP and AttnLRP in a grid-search (see Appendix B.5) resulted for both models in the same composite described in Table B.5. However, there is no assurance that other models share the same γ parameters.

Table B.5. Proposed composite for the AttnLRP and CP-LRP methods used for the Vision Transformer.

Layer Type	Rule Proposed
Convolution	Gamma($\gamma = 0.25$)
Linear	Gamma($\gamma = 0.05$)
LinearInputProjection	Epsilon
LinearOutputProjection	Epsilon

B.7. Attributions on SQuAD v2

In Figure B.7 and Figure B.8, we illustrate attributions on the Mixtral 8x7b for different state-of-the-art methods on the SQuAD v2 dataset. In Figure B.9, we depict attributions for Flan-T5-XL. For comparison, we also visualize a random attribution with Gaussian noise.

The similarity between AttnLRP and CP-LRP in Flan-T5-XL are in line with the quantitative evaluation from Table 1, which shows a small, but consistent advantage of AttnLRP over CP-LRP wrt. top-1 accuracy, while in Mixtral 8x7b and LLaMa 2, AttnLRP substantially outperforms, which is also visible in the heatmaps. This is due to the different number of non-linearities present in the models: Flan-T5-XL consists only of standard attention layers, while LLaMa

Figure B.6. Explanation heatmaps of the methods used for the perturbation experiments on the vision transformer ViT-B-16. A checkerboard effect is visible for almost every method, especially in AttnRoll (Abnar and Zuidema, 2020), G×AttnRoll (Chefer et al., 2021a), and AtMan (Deb et al., 2023). We improve upon CP-LRP (Ali et al., 2022) by applying the γ-rule as described in B.5. While qualitatively γCP-LRP (with γ extension for ViT) and AttnLRP give similar explanations, quantitative results in Table 1 and Table B.6 show a consistent improvement of AttnLRP over γCP-LRP in terms of faithfulness. Moreover, attributing query and key linear layers within the attention layer is possible with AttnLRP only, while it is not possible with CP-LRP. We leave these further explorations for future work.

Table B.6. Faithfulness scores as area between the least and most relevant order perturbation curves (Blücher et al., 2024) for ViT-L-16 and ViT-L-32 on ImageNet. For ViT-L-16, we set for SmoothGrad $\sigma = 0.01$, for AtMan $p = 1.0$ and $t = 0.1$, for AttnRoll $dt = 0.90$, and for G×AttnRoll $dt = 0.92$. For ViT-L-32, we set for SmoothGrad $\sigma = 0.01$, for AtMan $p = 1.0$ and $t = 0.1$, for AttnRoll $dt = 0.95$, and for G×AttnRoll $dt = 1.0$.

Method	**ViT-L-16** ImageNet ↑	**ViT-L-32** ImageNet ↑
Random	0.01 ± 0.01	0.01 ± 0.01
Input×Grad (Simonyan et al., 2014)	1.20 ± 0.06	0.98 ± 0.07
IG (Sundararajan et al., 2017)	0.96 ± 0.07	1.45 ± 0.06
SmoothGrad (Smilkov et al., 2017)	-0.10 ± 0.01	-0.09 ± 0.04
GradCAM (Chefer et al., 2021b)	0.19 ± 0.06	2.21 ± 0.08
AttnRoll (Abnar and Zuidema, 2020)	1.41 ± 0.08	1.90 ± 0.07
Grad×AttnRoll (Chefer et al., 2021a)	2.86 ± 0.06	2.69 ± 0.06
AtMan (Deb et al., 2023)	1.58 ± 0.08	0.09 ± 0.05
KernelSHAP (Lundberg and Lee, 2017)	4.35 ± 0.04	4.90 ± 0.03
CP-LRP (ε-rule, Ali et al. (2022))	4.96 ± 0.05	4.07 ± 0.05
CP-LRP (γ-rule for ViT, as proposed here)	6.97 ± 0.04	5.99 ± 0.04
AttnLRP (ours)	$\mathbf{7.17} \pm 0.04$	$\mathbf{6.06} \pm 0.04$

2 and Mixtral 8x7b have additional FFN layers with non-linear weighting or routing layers making the attribution process more difficult for CP-LRP. This effect is studied in Appendix B.4. In general, gradient-based methods such as G×I, SmoothGrad, IG and Grad-CAM are noisy and often not informative. Attention Rollout and Grad×Attention Rollout suffer from background noise. While the performance of AtMan is in some cases excellent as in Figure B.9, the method fails in other cases as in Figure B.7.

In Figure B.7 and B.8 for Mixtral 8x7b, most methods fail to highlight the correct answer tokens most strongly, except AttnLRP, confirming the quantitative evaluation from Table 1.

In Figure B.9, the heatmaps of I×G or Grad-CAM seem to be inverted, hence we experimented with inverting the attributions on a subset, however we did not notice improvement and applied the rules with their original definition. AtMan produces highly sparse attributions, assigning large positive relevance to the answer token `18`, however, also assigning a similar amount of relevance to the token `much`, which is part of the question. AttnLRP and CP-LRP identify the token `18` as being the most relevant token and also relate it (by assigning positive and negative relevance) to other information in the text such as `27.7`, `132` or `average`. We conjecture that such targeted contrasting reflects the reasoning process of the model (*e.g.*, is necessary to distinguish between related questions about how many tons are blown out vs. how many tons remain on the ground). A systematic analysis of these effects remain an interesting topic for future work.

B.8. Benchmarking Cost, Time and Memory Consumption

We benchmark the runtime and peak GPU memory consumption for computing a single attribution for LLaMa 2 with batch size 1 on a node with four A100-SXM4 40GB, 512 GB CPU RAM and 32 AMD EPYC 73F3 3.5 GHz. Because AtMan, LRP and AttnRollout-variants need access to the attention weights, we did not use flash-attention (Dao et al., 2022).

To calculate energy cost, we assume a price of $0.16\,\$$ per kWh of energy, and that a single A100 GPU consumes on average 130W. Figure B.10 depicts the cost, the runtime and peak GPU memory consumption. Since perturbation-based methods are memory efficient, a 70b model with full context size of 4096 is attributable. However, LRP with checkpointing requires more memory than a node supplies.

B.9. Attributions of Knowledge Neurons

Figure B.11, B.12 and B.13 illustrate the top 10 sentences in the Wikipedia summary dataset that maximally activate a knowledge neuron. We applied AttnLRP to highlight the tokens inside these reference samples. We observe that knowledge neurons exhibit remarkable disentanglement, *e.g.*, neuron `#256` of `layer 18` shown in Figure B.11 seems to encode concepts related to transport systems (railways in particular), while neuron `#2207` of `layer 20` shown in Figure B.12 seems to encode the concept teacher, in particular a teacher, in an unusual context (*e.g.*, inappropriate behavior, sexual misconduct). The degree of disentanglement should be studied in future work.

Question: In what country is Normandy located?
Answer: France

AttnLRP

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Input x Gradient

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

AtMan

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Grad-CAM

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Integrated Gradient

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

CP-LRP

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Grad-weighted Attention Rollout

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

SmoothGrad

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Attention Rollout

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Random

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Figure B.7. Evaluation on the Mixtral 8x7b model: We compute attributions for different state-of-the-art methods for the answer token "France". Gradient-based methods such as G×I, SmoothGrad, IG or Grad-CAM are noisy. Grad×Attn Rollout suffers from background noise. While AtMan usually generates sparse heatmap, in this case it fails (compare Figure B.9). CP-LRP highlights "Normandy" the strongest, while AttnLRP highlights the correct token "France". For comparison, we also visualize a random attribution with Gaussian noise.

Question: This person proposed explanations for the origins of earthquakes and the formation of mountains, what was his name?
Answer: Ibn Sina

Figure B.8. Evaluation on the Mixtral 8x7b model: We compute attributions for different state-of-the-art methods for all tokens at the same time inside the answer "Ibn Sina". Gradient-based methods such as G×I, SmoothGrad and IG are noisy. Grad-CAM highlights the correct tokens except it misses the beginning token "I" of the word "Ibn". Likewise AtMan fails to highlight all tokens. Grad×Attn Rollout suffers from background noise. CP-LRP resembles random noise, while AttnLRP highlights the correct tokens "Ibn Sina" in its entirety. For comparison, we also visualize a random attribution with Gaussian noise.

Question: How many tons of dust are blown out of the Sahara each year?
Answer: 182 million

AttnLRP

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Gradient x Input

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

AtMan

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Grad-CAM

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Integrated Gradient

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

CP-LRP

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Grad-weighted Attention Rollout

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

SmoothGrad

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Attention Rollout

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Random

Context: NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean (some dust falls into the Atlantic), then at 35 degrees West longitude at the eastern coast of South America, 27.7 million tons (15%) of dust fall over the Amazon basin, 132 million tons of dust remain in the air, 43 million tons of dust are windblown and falls on the Caribbean Sea, past 75 degrees west longitude. Question: How much dust is blown out of the Sahara each year? Answer:</s>

Figure B.9. Evaluation on the Flan-T5-XL model: We compute attributions for different state-of-the-art methods on the first token of the answer (highlighted in red). Gradient-based methods such as G×I, SmoothGrad, IG or Grad-CAM are noisy. Grad×Attn Rollout suffers from background noise. AtMan produces highly sparse attributions, assigning an equal amount of relevance to a token, which is part of the question, as to token `18`. CP-LRP has a different weighting of the tokens *e.g.* the word 'much' in the question is not highlighted by CP-LRP, while AttnLRP highlights it stronger and AtMan focuses excessively on it. For comparison, we also visualize a random attribution with Gaussian noise.

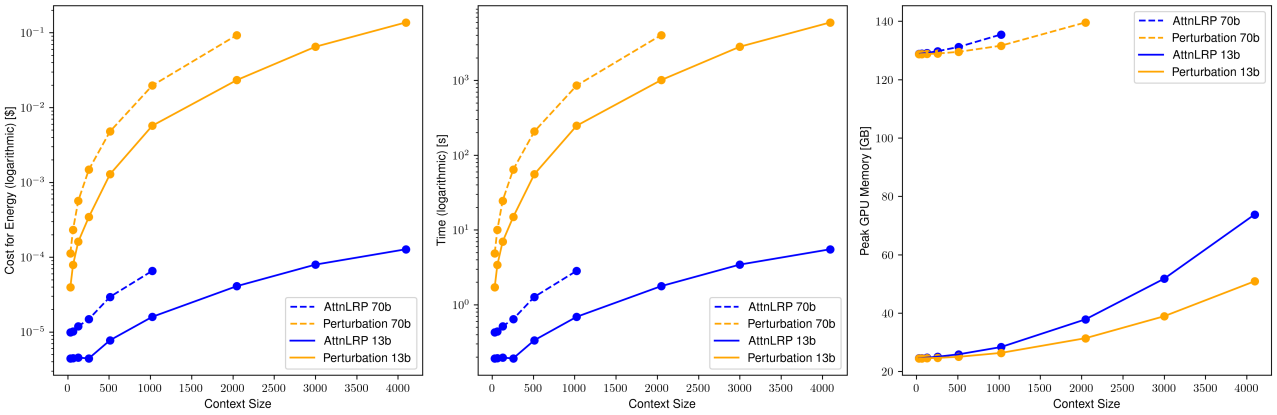

Figure B.10. From left to right: Cost in dollar, time in seconds and peak GPU memory in gigabytes for AttnLRP and linear-time perturbation. Evaluated on LLaMa 2-70b and LLaMa 2-13b models on a node with four A100-SXM4 40GB. G×AttnRollout is in the range of AttnLRP and omitted for clarity of visualization. Because AttnLRP consumes more than 160 GB of RAM, the curves for the 70b model stop. Measured at fixed intervals of context size 32, 64, 128, 256, 512, 1024, 2048, 3000, 4096.

Table B.7. ViT-B-16 Perturbation Experiment (Faithfulness). For SmoothGrad, we set $\sigma = 0.01$, for AtMan $p = 1.0$ and $t = 0.1$, for AttnRoll $dt = 0.99$, and for G×AttnRoll $dt = 0.91$. "all epsilon" indicates that the ε-rule has been used on the linear and convolutional layers. The term "best" refers to the utilization of LRP with the composite proposed in B.5. ΔA^F denotes the area under the curve for a *flipping* perturbation experiment which leverages both A^F_{MoRF} of the most relevant first order, and A^F_{LeRF} of least relevant first order. ($\Delta A^F = A^F_{LeRF} - A^F_{MoRF}$). As discussed in Section B.2, this is equivalent to *insertion* perturbation.

Methods	**ViT-B-16** ImageNet		
	$(\uparrow)\Delta A^F$	$(\downarrow)A^F_{MoRF}$	$(\uparrow)A^F_{LeRF}$
Random	0.01	4.71	4.71
I×G	0.90	2.78	3.69
IG	1.54	2.55	4.10
SmoothG	-0.04	3.63	3.58
GradCAM	0.27	5.35	5.63
AttnRoll	1.31	4.866	6.17
G×AttnRoll	2.60	4.01	6.22
AtMan	0.70	5.57	6.27
CP-LRP (all epsilon)	2.53	2.45	4.98
γCP-LRP (best)	6.06	1.53	7.59
AttnLRP (all epsilon)	2.79	5.22	2.42
AttnLRP (best)	**6.19**	**1.48**	**7.67**

Table B.8. Wikipedia Perturbation Experiment (Faithfulness). For SmoothGrad, we set $\sigma = 0.1$, for AtMan $p = 1.0$, for AttnRoll $dt = 1$, and for G×AttnRoll $dt = 1$. "all epsilon" indicates that the ε-rule has been used on all linear layers. ΔA^I denotes the area under the curve for the *insertion* perturbation experiment which leverages both A^I_{MoRF} of the most relevant first order, and A^I_{LeRF} of least relevant first order. ($\Delta A^I = A^I_{MoRF} - A^I_{LeRF}$). As discussed in Section B.2, this is equivalent to *flipping* perturbation.

Methods	**LLaMa 2-7b** Wikipedia		
	(↑)ΔA^I	(↑)A^I_{MoRF}	(↓)A^I_{LeRF}
Random	-0.07	2.31	2.38
I×G	0.18	1.27	1.09
IG	4.05	3.74	-0.31
SmoothG	-2.22	0.68	2.90
GradCAM	2.01	2.36	0.35
AttnRoll	-3.49	1.46	4.95
G×AttnRoll	9.79	8.79	-1.00
AtMan	3.31	4.06	0.76
CP-LRP (all epsilon)	7.85	6.43	-1.42
AttnLRP (all epsilon)	**10.93**	**9.08**	**-1.85**

Table B.9. IMDB Perturbation Experiment (Faithfulness), For SmoothGrad we set $\sigma = 0.05$, for AtMan $p = 0.7$, for AttnRoll $dt = 1$, and for G×AttnRoll $dt = 1$. "all epsilon" indicates that the ε-rule has been used to propagate relevance to the layers. ΔA^I demonstrates the area under the curve for the perturbation experiment of the type *Insertion* which leverages insights from both A^I_{MoRF} of the most relevant first order, and A^I_{LeRF} of least relevant first order. ($\Delta A^I = A^I_{MoRF} - A^I_{LeRF}$). As discussed in Section B.2, this is equivalent to *flipping* perturbation.

Methods	**LLaMa 2-7b** IMDB		
	(↑)ΔA^I	(↑)A^I_{MoRF}	(↓)A^I_{LeRF}
Random	-0.01	-0.47	-0.46
I×G	0.12	-0.69	-0.81
IG	1.23	-0.06	-1.29
SmoothG	0.25	-0.74	-0.98
GradCAM	-0.82	-1.10	-0.28
AttnRoll	-0.64	-0.64	0.00
G×AttnRoll	1.61	0.77	-0.84
AtMan	-0.05	-0.54	-0.49
CP-LRP (all epsilon)	1.72	0.50	-1.22
AttnLRP (all epsilon)	**2.50**	**1.12**	**-1.38**

The San Diego Electric Railway (SDERy) was a mass transit system in Southern California, United States, using 600 volt DC streetcars and (in later years) buses.
Maintenance of way (commonly abbreviated to MOW) refers to the maintenance, construction, and improvement of rail infrastructure, including tracks, ballast, grade, and lineside infrastructure such as signals and signs.
France currently operates the second-largest European railway network, with a total of 29,901 kilometres of railway.
The MHR had, in 1846, amalgamated with the "Little" North Western Railway (NWR), which was taken over by the Midland Railway in 1874.Awdry, p.97ĊThe rival London and North Western Railway (LNWR) built its own branch line to Morecambe in 1864, joining the main LNWR line at Hest Bank.
Some railway companies had a standard signalbox design, such as the London & North Western Railway, whereas others, such as the Great Eastern Railway had many different designs.
There are more than 16,000 student tour operators and travel agencies estimated in this market.
In 2010 it was totally integrated with the main regional public transport company, ARST (Azienda Regionale Sarda Trasporti).
A school was operating in the town in 1914.
The railway opened in 1886 with four stations using steam locomotives hauling unheated wooden carriages; in the next six years the line was extended and three more stations opened.
The MontrealJonqui train (formerly the Saguenay) is a passenger train operated by Via Rail between Montreal and Saguenay (borough of Jonqui) in Quebec, Canada.

Figure B.11. AttnLRP attributions on top 10 ActMax sentences collected over the Wikipedia summary dataset for neuron `#256`, in `layer 18`. The knowledge neuron seems to activate for transport systems (railways in particular).

The Court held unanimously in favor of a schoolteacher fired for her critical remarks in conversations with her principal.
The town schoolteacher was reading the book to her students when she was asked by her husband, the postmaster, to help name the little settlement.
She was a teacher for forty years and her writing has appeared in journals and anthologies since the early 1980s.
Her case made headlines and was covered by major news networks for being a notorious teacher who had an unlawful sexual relationship with one of her students.
His departure in 1971 generated some controversy on campus; he was regarded as an excellent teacher by his students, however, the administration was viewed as being more concerned about research than education when making its tenure decisions.
The volume presents six short stories, with the titular story featuring Yahiro, a substitute teacher, who begins having an affair with his student Kago.
In 2018, Derek Michael Boyce, a high school math and science teacher at the school, was arrested for having an inappropriate relationship with one of his students, a fifteen-year-old girl.
The film follows a school teacher as she suspects one of her students is suffering from personal problems in his home life, not knowing that the student is harboring an evil demon in his house.
During his time as a teacher Franco admitted to having sex with several of his students, which led to lawsuits and a $2 million sexual-misconduct settlement in 2021.
It tells the story of a schoolteacher who falls in love with one of his students, and moves away in order to escape his infatuation.

Figure B.12. AttnLRP attributions on top 10 ActMax sentences collected over the Wikipedia summary dataset for neuron `#2207`, in `layer 20`. The knowledge neuron is activating for 'teacher', in unusual context such as inappropriate behavior, sexual misconduct etc.

This contrasts with the pattern in all vascular plants (seed plants and pteridophytes), where the diploid sporophyte generation is dominant.
Like most other plants in the family, these produce umbels of flowers.Genus of the Month: Zizia.
Fructification () are the generative parts of the plant (flower and fruit) (as opposed to its vegetative parts: trunk, roots and leaves).
Symptoms of red stele can include a red core in the roots, wilting of leaves, reduced flowering, stunting, and bitter fruit.
Catopsis is a genus in the botanical family Bromeliaceae, subfamily Tillandsioideae.
Dracaena eilensis, synonym Sansevieria eilensis, is a xerophytic CAM succulent native to a small region of Somalia near the town of Eyl.
Dracaena suffruticosa, synonym Sansevieria suffruticosa is a species of Dracaena native to eastern Africa, from Ethiopia to Malawi.
Like other plant families, the Solanaceae is divided further into subfamilies, tribes and subtribes.
Pathogens that cause wilting diseases invade the vascular vessels and cause the xylem to fail to transport water to the foliage, thus causing wilting of stems and leaves.
Dracaena hanningtonii, synonym Sansevieria ehrenbergii, (blue sansevieria, sword sansevieria, oldupai, or East African wild sisal) is a flowering plant which grows in northeastern and eastern tropical Africa (Djibouti, Eritrea, Ethiopia, Kenya,Somalia, Sudan and Tanzania) and the Arabian Peninsula (Oman and Saudi Arabia).

Figure B.13. AttnLRP attributions on top 10 ActMax sentences collected over the Wikipedia summary dataset for neuron `#922`, in `layer 18`. The knowledge neuron seems to be activating for scientific descriptions of plants.

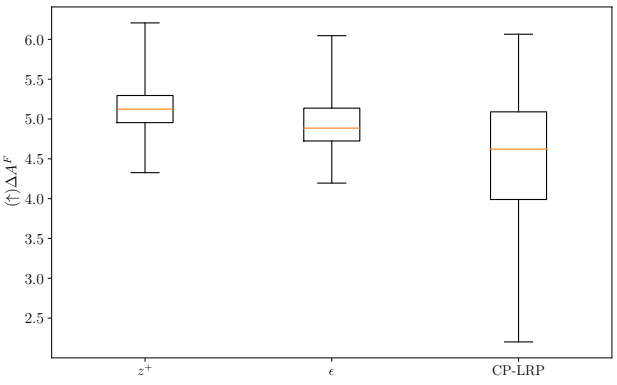

Figure B.14. Statistics on Rules used for *softmax* layers: Either applying z^+, ε-rule, or regarding as constant as proposed in CP-LRP. Propagating relevance values through (specifically by applying z^+ rule) *softmax* improves the faithfulness of explanations compared to the case where we block its propagation.

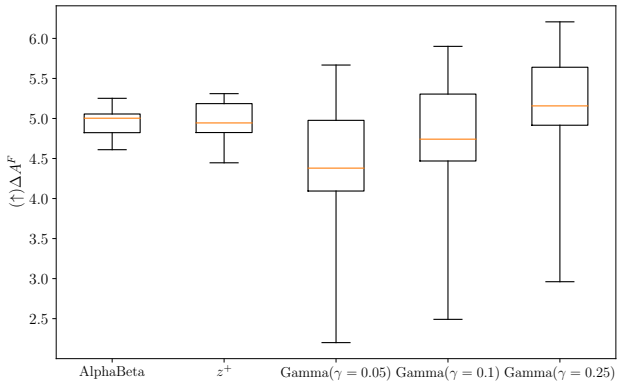

Figure B.15. Statistics on Rules used for *Convolution* layers: Applying z^+ and AlphaBeta proposes acceptable results however the most faithful results can be reached via Gamma($\gamma = 0.25$).

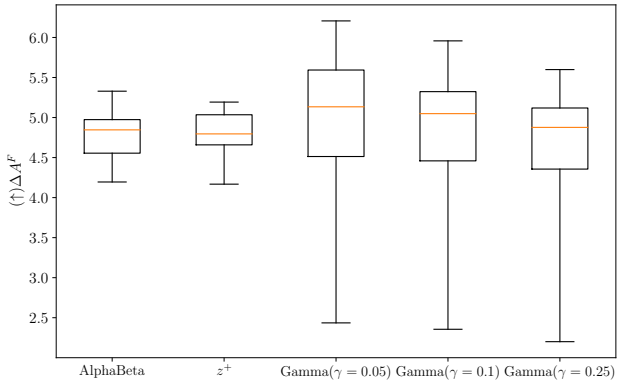

Figure B.16. Statistics on Rules used for *Linear* layers: Similar to *Convolution* layers, Gamma seems more promising however with different γ value (0.05 in this case).

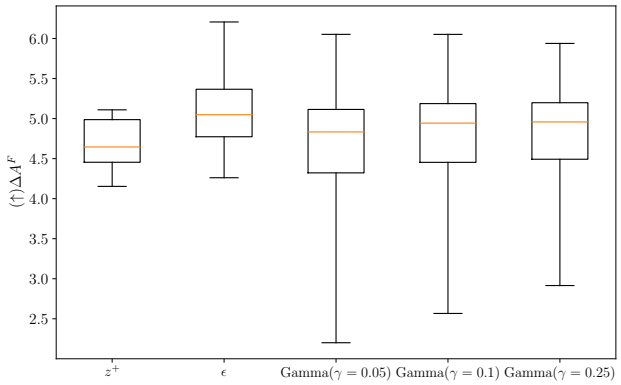

Figure B.17. Statistics on Rules used for *LinearInputProjection* layers: Gamma and ϵ rules are competitive in this case, however since there is larger difference between the minimum and the lower quartile in Gamma rules, the most faithful choice will be ϵ-rule.

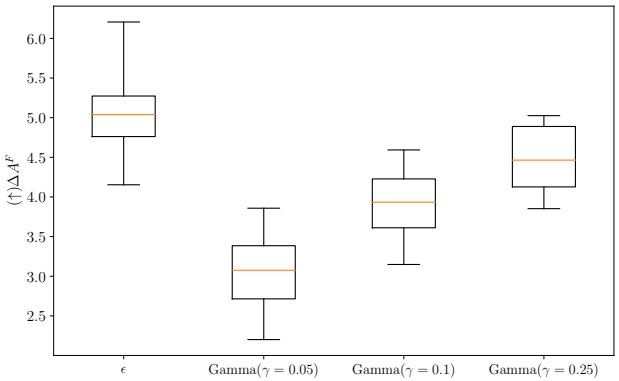

Figure B.18. Statistics on Rules used for *LinearOutputProjection* layers: The ϵ-rule outperforms other rules clearly.