A Close Look at Decomposition-based XAI-Methods for Transformer Language Models

Anonymous ACL submission

Abstract

Various XAI attribution methods have been proposed recently for the transformer architecture, allowing for insights into the decision-making process of large language models by assigning importance scores to input tokens and intermediate representations. One class of methods that seems very promising in this direction includes *decomposition*-based approaches, i.e., XAI methods that redistribute the model's prediction *logit* through the network, as this value is directly related to the prediction. In the previous literature we note though that two prominent methods of this category, namely ALTI-Logit and LRP, have not yet been analyzed in juxtaposition and hence we propose to close this gap by conducting a careful quantitative evaluation w.r.t. ground truth annotations on a subject-verb agreement task, as well as various qualitative inspections, using BERT, GPT-2 and LLaMA-3 as a testbed. Along the way we compare and extend the ALTI-Logit and LRP methods, including the recently proposed AttnLRP variant, from an algorithmic and implementation perspective. We further incorporate in our benchmark two widely-used gradient-based attribution techniques. Finally, we make our carefullly constructed benchmark dataset for evaluating attributions on language models, as well as our code[1], publicly available in order to foster evaluation of XAI methods on a well-defined common ground.

1 Introduction & Background

1.1 Interpretability of Transformers

Many approaches have been explored to shed light on how Transformer models process language. BERTology works (Rogers et al., 2020) primarily employ probes (Gupta et al., 2015; Köhn, 2015; Alain and Bengio, 2016) to analyze what information the model's internal representations encode,

[1]Link will be made available upon paper acceptance.

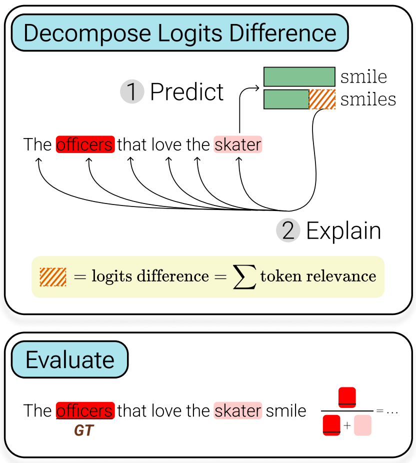

Figure 1: Our XAI evaluation pipeline using subject-verb agreement: 1) Predict the logits difference for the two verb forms, 2) Explain the logits difference by generating a token-level relevance heatmap for each XAI method (for decomposition-based XAI methods the relevances sum up to the logits difference), 3) Evaluate the heatmaps w.r.t. ground truth linguistic evidence (i.e., the subject) by computing various relevance accuracy metrics (such as the fraction of positive relevance falling inside the GT).

which can range from linguistic properties to factual and world knowledge (Clark et al., 2019; Hewitt and Manning, 2019; Liu et al., 2019; Petroni et al., 2019; Tenney et al., 2019; Cui et al., 2021, *inter alia*). However, probing itself is not without limitations, as it is correlational in nature and requires careful interpretation (Hewitt and Liang, 2019; Belinkov, 2022).

This spurs another line of inquiry asking how information is actually being used via causal intervention (Pearl, 2001; Vig et al., 2020; Geiger et al., 2021). Elazar et al. (2021) propose amnesic probing that ablates certain linguistic properties such as part-of-speech from models' representation to see how it affects actual predictions. Similarly, Meng et al. (2022) analyse how LLMs store and recall factual information by intervening on weights and hidden representations. Causal methods allow the isolation of subgraphs of neural networks that are responsible for certain tasks such as indirect object identification (Wang et al., 2023) and induction heads (Olsson et al., 2022), although the process itself relies on a non-trivial amount of manual labor. Recent efforts such as ACDC (Conmy et al., 2023) attempt to alleviate this issue, yet still can miss some nodes that are supposed to be part of the subgraph.

Our focus lies on attribution methods, specifically those that decompose the prediction logit throughout the entire network and do not only consider parts of the Transformer model such as MLPs (Geva et al., 2021, 2022) or attention modules (Abnar and Zuidema, 2020; Kobayashi et al., 2020). These approaches are able to measure causal properties (Geiger et al., 2021), allowing for the identification and localization of features playing an important role during inference in a more scalable manner, and simultaneously enable inspection of information encoded in the model (Achtibat et al., 2023; Ferrando and Voita, 2024).

1.2 Evaluation of Attributions

A common way to evaluate attributions is to systematically perturb parts of the models' inputs according to their relevance and then measure the resulting changes in the output, the higher the change the more accurate the attribution. Such an approach has been initially proposed as pixel-perturbation in the computer vision domain (Bach et al., 2015; Samek et al., 2017), and was later extended to words and tokens in NLP (Arras et al., 2016, 2019b; DeYoung et al., 2020).

Another direction is to use syntactic tasks to evaluate attributions, such as subject-verb agreement, since those tasks typically allow for the creation of ground truth annotations. Although such approach has been quite popular, to the best of our knowledge there exist no properly constructed and publicly available benchmark dataset using subject-verb agreement on real-world natural language data, and existing benchmarks often were constructed automatically by using short and simple sentence templates such as done for instance in BLiMP and CausalGym (Warstadt et al., 2020; Arora et al., 2024).

Besides automatic evaluation, user studies were also widely used to evaluate explanations (Doshi-Velez and Kim, 2017; Lipton, 2018; Hase and Bansal, 2020).

In the present work we contribute the following: (a) Analyze and compare decomposition-based attribution methods which were not yet compared to one another; (b) Generate and release a ground truth annotated real-world dataset for evaluating attributions on Language Models using a subject-verb agreement task; (c) Extend the ALTI-Logit decomposition-based XAI method to the Llama model family; (d) Propose a novel fast and simple method to implement the AttnLRP decomposition-based XAI method based on a *modified* Gradient×Input strategy, as well as provide a complete set of proofs to justify this approach for both XAI methods LRP and AttnLRP.

2 XAI Methods Strategy

Let us first introduce some notations that will help us analyze and compare the strategy of the considered XAI methods. Let x_t^l be the token representation for timestep t and layer l, and R_t^l the corresponding relevance[2] for this token. Accordingly R_t^0 represents the relevance of the input token for timestep t. Let $\mathbf{U}$ be the output embedding matrix, and U_w the column vector for the predicted token w. Hence, the language model's prediction logit for predicting token w at timestep T[3] is: $\text{logit}_w = x_T^L \cdot U_w$, with L being the number of layers of the model. A property which is common to all decomposition-based XAI methods is that the logit_w is decomposed additively into contributions of model components (token, neuron, head or layer), or in other words, the contributions of model components sum up to the value logit_w.

2.1 ALTI-Logit

ALTI-Logit is a recently proposed state-of-the-art decomposition-based approach for Transformer Language Models proposed by Ferrando et al. (2023). Its central idea is to additively decompose the final layer's token representation x_T^L used to compute the prediction logit (i.e., the penultimate vector ahead of the output embedding layer $\mathbf{U}$) into layer-wise contributions of the outputs of each MLP and MHA block[4], by following the resid-

[2] In this work we use the terms relevance, contribution, attribution and importance score interchangeably.

[3] Here we assume an Autoregressive Language Model, with T being the input length, but all considered XAI methods are in principle applicable to Masked Language Models as well.

[4] We refer to MLP as Multi-Layer Perceptron, and MHA as Multi-Head Attention, representing the two main components of the Transformer architecture.

ual connections of the model (Elhage et al., 2021). While the contributions of the MLP blocks are not decomposed further backward, the contributions of the MHA blocks get further broken down into contributions of their respective input token representations, similarly to attention decomposition from Kobayashi et al. (2021). The latter is achieved by linearizing the MHA-block by viewing the attention weight matrix as a constant, as well as treating the standard deviation within the normalization layers as a constant, similarly to how Layer-wise Relevance Propagation (LRP) was previously extended to Transformers (Ali et al., 2022). Lastly, in order to account for the mixing of information across multiple layers, a token-level contribution matrix is built within each MHA block by considering the contributions of the MHA's transformed vectors to the MHA's output vector (as was done in the ALTI method by Ferrando et al. (2022)), and the resulting matrices are multiplied across layers to finally obtain an ALTI-Logit contribution for each input token. Overall the decomposition property of ALTI-Logit can be summarized as follows[5]: $\sum_t R_t^0 + \sum_l \tilde{R}_T^l = \text{logit}_w$, where R_t^0 is the input token contribution for each timestep t in the input sequence resulting from the MHA blocks and aggregated over all layers, while $\tilde{R}_T^l$ is the contribution of the output of each MLP block for the given prediction timestep T and layer l, since ALTI-Logit assumes there is no mixing of information across timesteps resulting from MLP blocks.

In practice, the official implementation of ALTI-Logit[6] from (Ferrando et al., 2023) requires the computation of a second, carefully designed forward pass through the model (using attention matrices, as well as weight parameters from various intermediate layers), after having run a first standard forward pass through the model during which the inputs and outputs of hidden layers are collected via hooks. This dedicated forward pass in ALTI-Logit was so far derived for Pre-LayerNorm architectures[7] and Autoregressive Language Models, and exclusively applied to the models GPT2 (Radford et al., 2019), OPT (Zhang et al., 2022) and BLOOM (Scao et al., 2023).

[5]Ignoring contributions from model biases for simplicity of notation.

[6]https://github.com/mt-upc/logit-explanations

[7]We refer to *Pre*-LayerNorm to indicate that the normalization layer is located *before* the self-attention computation (resp. the fully-connected layers) within the MHA (resp. MLP) blocks, as opposed to *Post*-LayerNorm where the normalization happens *after* them.

In this work we extend the ALTI-Logit algorithm to the Llama model family (Touvron et al., 2023; Grattafiori et al., 2024) by adapting ALTI-Logit to handle grouped-query attention (Ainslie et al., 2023), as well as RMSNorm normalization (Zhang and Sennrich, 2019). However, we refrain from adapting ALTI-Logit to the BERT model family, as this would require a substantial re-design of the algorithm to cope with Post-LayerNorm architectures as well as Masked Language Modeling.

ALTI-Logit provides layer-wise token-level (as well as head-level) contributions to the prediction logit, and this method (resp. its components Logit (Ferrando et al., 2023) and ALTI (Ferrando et al., 2022)) were previously evaluated against Erasure (Li et al., 2017), Gradient (Simonyan et al., 2014; Li et al., 2016), Gradient×Input (Denil et al., 2015; Shrikumar et al., 2016), Integrated Gradients (Sundararajan et al., 2017), Attention Rollout (Abnar and Zuidema, 2020) and GlobEnc (Modarressi et al., 2022) explanations, where ALTI-Logit was shown to deliver the best results.

2.2 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) is an interpretability method based on backward decomposition following a layer-wise conservation principle. In other words, in each layer of the model the contributions of neurons sum up to the prediction logit. More precisely it holds[8]: $\sum_t R_t^0 = \sum_t R_t^1 = \cdots = \sum_t R_t^L = \text{logit}_w$. LRP was initially proposed for Convolutional Neural Networks (Bach et al., 2015), and later extended to other models such as Recurrent Networks (Arras et al., 2017, 2019a), Transformers (Ali et al., 2022; Achtibat et al., 2024) and selective State Space Models (Rezaei Jafari et al., 2024).

In practice, LRP can be implemented by applying dedicated LRP backward propagation rules for each type of layer occuring in the network, and that redistribute neuron relevances from upper layers to lowers layers in a conservative manner (Montavon et al., 2019).

For a linear layer with forward pass equation $z_j = \sum_i z_i w_{ij} + b_j$, and given the relevances of the output neurons R_j, the input neurons' relevances R_i are computed through a summation of the form[9]:

[8]Here also ignoring the relevances assigned to model biases for simplicity.

[9]This rule corresponds to the LRP-ϵ rule (with ϵ being a small numerical stabilizer) which was shown to work well in NLP. On computer vision models, in particular for convolu-

$R_i = \sum_j \frac{z_i \cdot w_{ij}}{z_j + \epsilon \cdot \text{sign}(z_j)} \cdot R_j$, hence their relevances are proportional to their forward pass contributions. For an element-wise activation layer of the form $z_j = g(z_i)$, with g being a non-linear activation function, the relevance R_j is redistributed backward using the identity rule, thus $R_i = R_j$. In order to extend LRP to Transformer models, it is required to design new rules to propagate the relevance backward through two further non-linearities typical to the models' architecture: product layers (occurring for instance in the product between attention weights and value vectors inside the MHA), and the normalization layer (LayerNorm or RMSNorm). To this end Ali et al. (2022) propose to view the attention weights as a constant, which is equivalent to using the signal-take-all LRP redistribution rule for products which was previously proposed for extending LRP to Recurrent Neural Networks (Arras et al., 2017, 2019a). For the normalization layers, Ali et al. (2022) propose to treat the standard deviation as a constant. In practice, these two rules can be implemented by treating the previous non-linearities as linear layers for LRP (see Appendix D for more details).

While the LRP extension to Transformers has been proposed in Ali et al. (2022), early implementations of LRP on Transformers Ali et al. (2022); Eberle et al. (2022) omit the redistribution of relevance through MLP blocks (more particularly through its element-wise activation layer), and were only utilizing LRP rules inside MHA blocks. To the best of our knowledge the first LRP implementation applied to a complete Transformer architecture was provided by Eberle et al. (2023). Ali et al. (2022) evaluated LRP against various attention-based XAI methods (Abnar and Zuidema, 2020; Sood et al., 2020; Chefer et al., 2021a), as well as Gradient×Input (Denil et al., 2015; Shrikumar et al., 2016), and LRP was shown to deliver the best results.

2.3 AttnLRP

AttnLRP is a novel variant of LRP (Achtibat et al., 2024), which in contrast to ALTI-Logit and LRP does not consider the attention weights as a constant, and thus redistributes relevances backward onto the key and query vectors. In particular Achtibat et al. (2024) handles product layers by employing "uniform" LRP redistribution rule. Concretely, given a product layer $z_a \cdot z_b = z_j$, the relevance of the output neuron R_j is redistributed equally among input neurons, hence $R_a = R_b = 0.5 \cdot R_j$. This is similar to a rule previously proposed for extending LRP to customized LSTMs (Arras et al., 2019a; Arjona-Medina et al., 2019). As a result, the attention weights' matrix is assigned relevance scores, opening up the question of how to redistribute this quantity further backward through the softmax non-linearity. For that purpose Achtibat et al. (2024) propose a novel redistribution rule which is equivalent to using the Gradient×Input XAI method for that layer. While this redistribution strategy does not conserve the overall relevance between the layer's output and input neurons, it can be justified by the fact that during the forward pass the softmax layer may have a non-zero output while all inputs are zero, which can be interpreted as a bias parameter for that layer[10].

Currently, an implementation of AttnLRP (Achtibat et al., 2024) is available via the highly specialized LXT[11] toolbox, which overwrites the Pytorch backward function of all layers present in the network. In Section 2.7, we will show that a strategy similar to the one previously adopted for LRP based on a modified Gradient×Input approach can also be extended to AttnLRP to allow for a simpler and faster implementation. AttnLRP was evaluated against LRP from Ali et al. (2022), as well as various attention-based (Abnar and Zuidema, 2020; Chefer et al., 2021a,b; Deiseroth et al., 2023) and gradient-based (Simonyan et al., 2014; Sundararajan et al., 2017; Smilkov et al., 2017) XAI methods, and AttnLRP was shown to deliver the best results.

2.4 Gradient-based

We consider two gradient-based methods commonly used in previous XAI works. Both approaches compute the gradient of the prediction logit w.r.t. the input token's representation of interest and normalize it using either the L_1-norm or squared L_2-norm, i.e., $R_t^0 = \|\nabla_{x_t^0} \text{logit}_w\|_1$,

tional layers, other rules have be shown to be more adequate (Montavon et al., 2019; Arras et al., 2022; Kohlbrenner et al., 2020)

[10] This is similar to how biases in linear layers get assigned (or absorb) a portion of the relevance. Indeed, strictly speaking, with LRP the sum of the input tokens' relevances will be numerically equal to the prediction logit only if all model biases are set to zero (which in practice can serve as a sanity check for the LRP implementation). See the redistribution rule for linear layers introduced in Section 2.2, where the bias term appears in the denominator.

[11] https://github.com/rachtibat/LRP-eXplains-Transformers/tree/25aa8f3 (latest available commit at the time of submission: Feb 13th 2025, 25aa8f3)

resp. $\|\nabla_{x_t^0} \text{logit}_w\|_2^2$. Both variants have the advantage of being on an additive scale, meaning that the contributions of smaller units (neurons, tokens, or words) can be summed up to obtain the relevance of a greater portion of the input. We tried both and report only the best results under Gradient. The Gradient×Input method computes the dot product between the gradient and the input token's representation, i.e. $R_t^0 = \nabla_{x_t^0} \cdot x_t^0$. All gradient-based methods are easy and efficient to compute, and can be obtained via standard gradient backpropagation.

2.5 Overview

Table 1 summarizes all XAI attribution methods considered in this work, whereas only the first three methods ALTI-Logit, LRP and AttnLRP are decomposition-based and redistribute the prediction logit's quantity onto model components at different levels of granularity. While ALTI-Logit assigns relevance at the token-level, and if desired also at head-level inside MHA blocks, LRP and AttnLRP are more fine-grained methods and decompose the prediction down to the smallest possible unit, i.e., a neuron. Regarding computation time, all methods have conceptually a similar cost in number of forward/backward passes required, though depending on the efficiency of the particular implementation that is used different memory and time costs might arise in practice (as we will see for instance for AttnLRP in Section 2.7). In order to additively decompose the prediction logit into contributions of model components, decomposition-based XAI methods make several simplifying assumptions: in particular they tend to "linearize" parts of the model (e.g., by viewing the attention matrix as a constant, or treating the standard deviation inside normalization layers as a constant, see Appendix D for more details). Gradient-based explanations do not make those simplifications, though they are unable to explain the actual prediction's logit, but explain instead its derivatives (or in other words, per definition, they identify tokens/neurons of which a slight perturbation might influence a significant change in the prediction). Finally, while most XAI methods redistribute relevances backward across all layers of the model, and thereby take into account a mixing of contextual information arising from token-interactions inside MHA block, ALTI-Logit is the only method where the flow of information gets truncated inside MLP blocks and is not backward propagated further from these layers on (except for contributions from residual connections).

2.6 Methods not considered

Other non-decomposition based XAI methods which we do not consider include more sophisticated gradient-based variants such as Integrated Gradients (Sundararajan et al., 2017) and SmoothGrad (Smilkov et al., 2017). These methods try to alleviate the noisy gradient problem (Balduzzi et al., 2017) by averaging gradients over several perturbed samples. However they introduce hyperparameters into the explanation process (such as the number/type of perturbations or the baseline choice[12]), and in a typical XAI use-case (with no available ground truth) one has no criteria to tune those hyperparameters. Further, similarly to perturbation-based XAI methods, generating and leveraging the perturbations yields an additional computation cost (one typically needs one backward, resp. forward pass, for each perturbed sample with gradient-based, resp. perturbation-based, XAI methods). Other popular non-decomposition based XAI methods include attention-based methods such as Attention Rollout (Abnar and Zuidema, 2020) and ALTI (Ferrando et al., 2022). Although these methods are intuitively appealing since they leverage the mixing of information already provided by attention weights and trace it back across layers, those methods have been shown to be inferior to decomposition-based methods in previous works (Ali et al., 2022; Achtibat et al., 2024; Ferrando et al., 2022, 2023), and are typically not directly related to a specific class/token prediction. Lastly we do not consider other LRP-based approaches for Transformers proposed in the literature (Voita et al., 2021; Chefer et al., 2021b): those do not follow a layer-wise conservation principle within the MHA layer[13] and have been observed to lead to numerical instabilities (Achtibat et al., 2024).

2.7 LRPx : Fast and simple implementation of LRP variants

The adoption of LRP (and its variant AttnLRP) has been so far mainly tied to the use of ready-made and highly specialized toolboxes (such as Zennit (Anders et al., 2021), LXT (Achtibat et al., 2024) or

[12]Indeed these hyperparameters can have a huge impact on the quality of explanations, as was previously shown in computer vision (Arras et al., 2022), for instance a zero-valued baseline as is often used for the Integrated Gradients method might be sub-optimal.

[13]In fact they enforce conservation artificially via a subsequent normalization step over relevances.

Table 1: Overview of the XAI attribution methods considered in this work.

Method	granularity	computation	treat normalization as a linear layer	treat attention matrix as a constant	mixing of information upward MLP blocks	logit decomposition
ALTI-Logit	token, head, layer	2 × forward	✓	✓	✗	✓
LRP	neuron, layer	forward + backward	✓	✓	✓	✓
AttnLRP	neuron, layer	forward + backward	✓	✗	✓	✓
Gradient, Gradient×Input	neuron, layer	forward + backward	✗	✗	✓	✗

others (Lapuschkin et al., 2016; Alber et al., 2019)). Such toolboxes compute the LRP relevances explicitly at each layer by overwriting the standard gradient backward pass (either through hooks, and by overwriting the backward function of every layer). However, it is possible to implement LRP on Transformers in a more lightweight and elegant manner by adopting a *modified* Gradient×Input strategy. To the best of our knowledge the first work where this strategy was employed was Eberle et al. (2023). It consists in modifying a few layers during the forward pass (only non-linear layers need to be modified, so far less layers than in Zennit or LXT) such that their output values remain unchanged (hence without affecting the forward pass outcome), but in a way that the resulting gradients from the Pytorch´s automatic differentiation engine multiplied with the forward pass activations yields LRP relevances at any hidden or input layer of interest (in practice this is achieved by detaching dedicated neurons from the computational graph by using Pytorch's `Tensor.detach()` method). Although this efficient and simple strategy to implement LRP has been further adopted in a recent work extending LRP to State Space Models (Rezaei Jafari et al., 2024), and builds upon various LRP properties and derivations provided in multiple previous works (Lapuschkin, 2019; Eberle, 2022; Montavon et al., 2019; Rezaei Jafari et al., 2024), to the best of our knowledge there exist so far no comprehensive and complete set of proofs demonstrating the equivalence of explicit LRP rules with this *modified* Gradient×Input approach. In the present work we close this gap by providing such extensive proofs in the Appendix D.

Further, we show for the first time that the *modified* Gradient×Input strategy can also be extended to AttnLRP. As mentioned earlier, AttnLRP differs from LRP by the rules it employs for the product and softmax layers, see Section 2.3. Let us consider the modified product layer defined by: $\hat{z}_j = 0.5 \cdot (z_a \cdot z_b) + [0.5 \cdot (z_a \cdot z_b)]_{\text{detach()}}$. One can easily see that the forward pass outcome remains unchanged (i.e., $z_j = \hat{z}_j$). The resulting gradient of z_a is: $dz_a = 0.5 \cdot z_b \cdot dz_j$. Now let's assume relevances are computed via a Gradient×Input formula using this modified product layer, thus $R_j = dz_j \cdot z_j$ and $R_a = dz_a \cdot z_a$. As a result it holds: $R_a = 0.5 \cdot z_b \cdot dz_j \cdot z_a = 0.5 \cdot z_j \cdot dz_j = 0.5 \cdot R_j$, which is equivalent to the uniform rule for products $\square$. Hence we have shown that the uniform rule used in AttnLRP can be implemented via a *modified* Gradient×Input strategy. In the Appendix D.5 we provide a further proof that the AttnLRP redistribution rule for softmax proposed by Achtibat et al. (2024) is equivalent to Gradient×Input.

In this work we implement a straightforward and compact Pytorch toolbox named LRPx (where x stands for multiple LRP variants) which is part of our released code, and that allows to compute both LRP and AttnLRP using the *modified* Gradient×Input strategy. In Section 4.3 we benchmark the resulting computational time on Transformers using LRPx against the LXT toolbox from Achtibat et al. (2024).

3 A Benchmark for Language Model Attributions

3.1 Suject-Verb Agreement (SVA) Task

We build our XAI benchmark dataset for Language Models on top of the natural language subject-verb agreement dataset released by Goldberg (2019), which itself is based upon data from Linzen et al. (2016). In order to identify the subject of a given verb we employ Spacy's dependency parser and make sure that the dependency relation between the verb and the subject is of type "nominal subject". Note that previous work by Ferrando et al. (2023) also created a similar benchmark, however their ground truth subjects were incorrect[14]. We

[14]Indeed they used the first subject occuring in the sentence as ground truth, although it might not be in a dependency relation with the verb of interest in case of multi-phrase sentences. This bug has a huge impact on the results, e.g., on `GPT2-small` Ferrando et al. (2023) report a MRR accuracy of approx. 0.60 for the ALTI-Logit method, while we find 0.81.

build our dataset meticulously, additionally discarding some invalid and trivial samples, in order to release a proper and well-defined dataset to the research community. Appendix B provides all the details of the data generation process. Our resulting tokenized datasets contain 29k samples.

In order to explain the model's SVA predictions, we generate contrastive explanations (Yin and Neubig, 2022), in other words, we explain a logits difference of the form: $\text{logit}_p - \text{logit}_o$, where p indicates the predicted verb number (singular/plural) and o the opposite verb number.

3.2 Language Models

We employ the following models: `bert-base-uncased`, `bert-large-uncased`, `GPT2-small`, `GPT2-XL`, `Llama-3.2-1B` and `Llama-3.2-3B` from the HuggingFace library. Appendix Table 3 provides the models' prediction accuracy on SVA, as well as various informations on the models' sizes and tokenizer.

3.3 Evaluation Metrics

We employ four different evaluation metrics.

Pointing Game top-k (PGk). This metric looks at the top-k tokens with the highest relevances. If one of these tokens is within the ground truth, the accuracy is 1 else 0. We report results for k=2 in our experiments. A similar metric has been previously used to evaluate attributions (Poerner et al., 2018).

Mean Reciprocal Rank (MRR). This is the sole metric reported in the evaluation work by Ferrando et al. (2023). It consists in retrieving the inverse of the minimal rank (in decreasing order of relevance) of the tokens belonging to the ground truth.

Relevance Mass Accuracy (RMA). This metric was introduced in computer vision (Arras et al., 2022), and calculates the fraction of positive relevance that falls inside the ground truth over the total positive relevance present in the input.

Per-Token Accuracy (PTA). This metric makes a binary classification decision based on the sign of the relevance and then computes the classification accuracy w.r.t. the ground truth tokens. More precisely, it assumes tokens inside the ground truth shall receive a strictly positive relevance, while tokens outside the ground truth shall have no relevance or a negative relevance. It is related to the Pixel Accuracy used to evaluate semantic segmentation in computer vision.

4 Results

4.1 Evaluation w.r.t. Ground Truth

Table 2 presents our results. We computed the metrics using only correctly predicted samples. When we look at the PTA results, we find that Gradient has the best performance, and results are consistent across models. However, the score achieved by Gradient for this metric is close to random, and worse than random for the other XAI methods. This illustrates the importance of the choice of the metric for XAI evaluation, and reveals that the underlying assumption of PTA that the sign of the relevance shall switch between tokens inside and outside the ground truth is not adequate.

The remaining evaluation metrics deliver largely consistent result per Language Model family. While AttnLRP performs well on BERT and Llama3 models, ALTI-Logit is the strongest method on GPT2, followed by LRP. We are not yet able to understand why the results are so different across model families, since most components within the given architectures are similar. However we believe this constitutes an interesting finding that should be investigated in future work. In terms of magnitude of the metrics, results are higher for GPT2 and Llama3 than on BERT, which is probably due to the different tokenizers and vocabulary sizes that lead to longer inputs for BERT models, since this difference is also reflected in the random baseline results.

4.2 Exemplary Heatmaps

In the Appendix Fig. D.5 we provide some exemplary heatmaps using the five samples with the highest logits difference across the dataset for the model `Llama-3.2-1B`. One can see that heatmaps for AttnLRP are more sparse and focused than those for ALTI-Logit.

4.3 Computational Speedup with LRPx

We calculated the computational time speedup obtained for AttnLRP using our LRPx toolbox (i.e., a Gradient×Input strategy), versus the original LXT toolbox from Achtibat et al. (2024) (based on an explicit relevance computation). To this end, we retrieve the median speedup over the first 1,000 samples of our SVA dataset, using two types of GPUs: NVIDIA Tesla V100 32GB and NVIDIA A100 40GB, and single precision. On `bert-base-uncased` we obtained a speedup between 1.83 and 1.98, and on `Llama-3.2-1B` be-

Table 2: Token-level accuracy of the XAI methods w.r.t. ground truth, using different metrics (PG2: Pointing Game top-2, MRR: Mean Reciprocal Rank, RMA: Relevance Mass Accuracy, PTA: Per-Token Accuracy). All metrics are within [0.0, 1.0], the higher the better, we highlight in bold the best result, and underline the second best per model. The random baseline was obtained by sampling relevances uniformly in the range [-1.0, 1.0) for each given model's tokenized dataset (averaged over 10 runs).

	BERT				GPT2				Llama3			
	`bert-base-uncased`				`GPT2-small`				`Llama-3.2-1B`			
XAI Method	PG2↑	MRR↑	RMA↑	PTA↑	PG2	MRR	RMA	PTA	PG2	MRR	RMA	PTA
ALTI-Logit	-	-	-	-	**0.867**	**0.808**	<u>0.342</u>	0.330	<u>0.690</u>	<u>0.623</u>	<u>0.288</u>	0.359
LRP	0.688	0.637	0.208	0.339	<u>0.738</u>	<u>0.706</u>	**0.367**	<u>0.445</u>	<u>0.690</u>	0.533	0.221	0.244
AttnLRP	**0.775**	<u>0.705</u>	**0.260**	<u>0.376</u>	0.705	0.634	0.318	0.408	**0.884**	**0.814**	**0.387**	<u>0.382</u>
Gradient	**0.775**	**0.718**	<u>0.245</u>	0.041	0.592	0.551	0.255	0.153	0.283	0.366	0.151	0.127
Gradient×Input	0.292	0.316	0.095	**0.497**	0.262	0.353	0.152	**0.565**	0.365	0.407	0.183	**0.512**
	`bert-large-uncased`				`GPT2-XL`				`Llama-3.2-3B`			
ALTI-Logit	-	-	-	-	**0.885**	**0.852**	<u>0.402</u>	0.320	0.614	0.557	<u>0.266</u>	0.297
LRP	<u>0.560</u>	0.521	<u>0.187</u>	<u>0.408</u>	<u>0.823</u>	<u>0.754</u>	**0.407**	<u>0.392</u>	<u>0.747</u>	<u>0.622</u>	0.241	0.231
AttnLRP	**0.641**	**0.580**	**0.224**	0.383	0.779	0.658	0.351	0.384	**0.885**	**0.764**	**0.364**	<u>0.326</u>
Gradient	0.555	<u>0.551</u>	0.177	0.041	0.579	0.546	0.265	0.153	0.275	0.311	0.143	0.127
Gradient×Input	0.212	0.249	0.077	**0.513**	0.321	0.408	0.198	**0.608**	0.377	0.413	0.192	**0.514**
Random mean	0.080	0.149	0.040	0.500	0.277	0.360	0.151	0.501	0.241	0.326	0.127	0.501
±std	0.002	0.001	0.000	0.001	0.004	0.002	0.001	0.001	0.002	0.001	0.001	0.000

tween 1.54 and 1.61. Hence this illlustrates that the Gradient×Input approach for implementing LRP/AttnLRP is not only conceptually simpler, but also faster.

5 Outlook

While in this work we have focused on evaluating decomposition-based attributions on the input tokens, since for the input tokens one can easily define a ground truth, the relevances obtained for hidden layers might in principle also be useful to perform other tasks than merely explain the predictions, e.g., to unbias or improve the model's performance (Weber et al., 2023), increase model robustness to perturbations (Sun et al., 2025), or to prune and quantize the model (Yeom et al., 2021; Becking et al., 2022). Recently it has been shown that gradient-based relevances can be used in place of costly causal attribution methods to localize and control model behaviors to components (Kramár et al., 2024). The decomposition-based approaches discussed in this work might perform even better in this regard, since their token-level accuracies are generally higher than those of gradient-based methods. Another complementary direction to the present approach would be to consider synthetic tasks instead of natural language to evaluate XAI, in order to allow for a better control over biases and Clever Hans behaviors (Lapuschkin et al., 2019), or to use white-box models (Hao, 2020). Lastly, our evaluation approach using subject-verb agreement can also be extended to other Language Model architectures as well such as State Space Models or xLSTMs.

6 Conclusion

In this work we took a close look at state-of-the-art decomposition-based attributions, by analyzing their common characteristics as well as their differences. Further, we showed that the LRP-based explanations can be computed in a simple and fast way by using a *modified* Gradient×Input strategy. Our careful evaluation w.r.t. automatically generated ground truth annotations reveals that the quality of explanations differs across model families. Identifying the root causes for these differences shall constitute a topic for future work.

Limitations

Our ground truth data is automatically generated via using the dependency parser Spacy. Such a a toolbox is not 100% accurate, and hence might introduce some noise in the evaluation process. Further our benchmark dataset is extracted from real-world natural language data, and as such might contain misspellings, typos, and even grammatically incorrect sentences. However our goal in this work is to evaluate in a realistic setup, and we believe those limitations do not influence the comparison of XAI methods in a noticeable way.

References

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

Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. 2023. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019.

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. 2024. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 135–168. PMLR.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *Preprint*, arXiv:1610.01644.

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2019. innvestigate neural networks! *Journal of Machine Learning Research*, 20(93):1–8.

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. XAI for transformers: Better explanations through conservative propagation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 435–451. PMLR.

Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. 2021. Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *CoRR*, abs/2106.13200.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663, Bangkok, Thailand. Association for Computational Linguistics.

Leila Arras, José Arjona-Medina, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter, and Wojciech Samek. 2019a. Explaining and Interpreting LSTMs. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 211–238.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019b. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy. Association for Computational Linguistics.

Leila Arras, Ahmed Osman, and Wojciech Samek. 2022. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

David Balduzzi, Marcus Frean, Lennox Leary, J. P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. 2017. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 342–350. PMLR.

Daniel Becking, Maximilian Dreyer, Wojciech Samek, Karsten Müller, and Sebastian Lapuschkin. 2022.

ECQx: Explainability-Driven Quantization for Low-Bit and Sparse DNNs, pages 271–296. Springer International Publishing, Cham.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Hila Chefer, Shir Gur, and Lior Wolf. 2021a. Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 387–396, Los Alamitos, CA, USA. IEEE Computer Society.

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

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 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, Florence, Italy. Association for Computational Linguistics.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

Leyang Cui, Sijie Cheng, Yu Wu, and Yue Zhang. 2021. On commonsense cues in BERT for solving commonsense tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 683–693, Online. Association for Computational Linguistics.

Björn Deiseroth, Mayukh Deb, Samuel Weinbach, Manuel Brack, Patrick Schramowski, and Kristian Kersting. 2023. Atman: Understanding transformer predictions through memory efficient attention manipulation. In *Advances in Neural Information Processing Systems*, volume 36, pages 63437–63460. Curran Associates, Inc.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2015. Extraction of Salient Sentences from Labelled Documents. *arXiv:1412.6815*. Version 2.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *Preprint*, arXiv:1702.08608.

Oliver Eberle. 2022. *Explainable structured machine learning*. Phd thesis, Technische Universität Berlin. Https://doi.org/10.14279/depositonce-16149.

Oliver Eberle, Stephanie Brandl, Jonas Pilot, and Anders Søgaard. 2022. Do transformer models show similar attention patterns to task-specific human gaze? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4295–4309, Dublin, Ireland. Association for Computational Linguistics.

Oliver Eberle, Ilias Chalkidis, Laura Cabello, and Stephanie Brandl. 2023. Rather a nurse than a physician - contrastive explanations under investigation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6907–6920, Singapore. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *CoRR arXiv:1901.05287*.

Aaron Grattafiori et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Yiding Hao. 2020. Evaluating attribution methods using white-box LSTMs. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 300–313, Online. Association for Computational Linguistics.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

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

Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. 2020. Towards best practice in explaining neural network decisions with lrp. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Arne Köhn. 2015. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. *Preprint*, arXiv:2403.00745.

Sebastian Lapuschkin. 2019. *Opening the machine learning black box with Layer-wise Relevance Propagation*. Phd thesis, Technische Universität Berlin. Http://dx.doi.org/10.14279/depositonce-7942.

Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. The lrp toolbox for artificial neural networks. *Journal of Machine Learning Research*, 17(114):1–5.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10:1096.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding neural networks through representation erasure. *Preprint*, arXiv:1612.08220.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Zachary C. Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. Layer-wise relevance propagation: An overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 193–209. Springer.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Melbourne, Australia. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Farnoush Rezaei Jafari, Grégoire Montavon, Klaus-Robert Müller, and Oliver Eberle. 2024. Mambalrp: Explaining selective state space sequence models. In *Advances in Neural Information Processing Systems*, volume 37, pages 118540–118570. Curran Associates, Inc.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.

Teven Le Scao et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100. BigScience Workshop.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Interpretable deep learning by propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *International Conference on Learning Representations - Workshop track (ICLR)*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. In *Proceedings of the International Conference on Machine Learning Workshop on Visualization for Deep Learning*.

Ekta Sood, Simon Tannert, Diego Frassinelli, Andreas Bulling, and Ngoc Thang Vu. 2020. Interpreting attention models with human visual attention in machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 12–25, Online. Association for Computational Linguistics.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025. Transformer layers as painters. *Preprint*, arXiv:2407.09298.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3319–3328.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leander Weber, Sebastian Lapuschkin, Alexander Binder, and Wojciech Samek. 2023. Beyond explaining: Opportunities and challenges of xai-based model improvement. *Information Fusion*, 92:154–176.

Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2021. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

A Software Requirements & Licenses

All our experiments are conducted using the following python packages and their respective version numbers within a Python 3.11.9 environment:

- Spacy 3.7.3
- Pandas 2.2.1
- Pytorch 2.3.0
- Numpy 1.26.4
- HuggingFace Transformers 4.48.1

Licenses BERT is released under Apache 2.0, GPT-2 under MIT, and Llama-3 under Meta Llama 3 Community License.

B XAI Benchmark Dataset Generation

We build our XAI benchmark for language models on top of the natural language subject-verb agreement dataset released by Goldberg (2019) (available under: `https://github.com/yoavg/bert-syntax/blob/master/lgd_dataset.tsv`), which itself is based upon data from Linzen et al. (2016) with MIT license. This dataset is made of initially 29,985 uncased sentences from Wikipedia, each containing a verb in present tense, and allowing for a bidirectional stimuli with input beyond the verb's position in the sentence (i.e., for BERT-like masked language models). For causal language models (i.e., GPT2-like language models) we use as a stimuli only the portion of the sentence *before* the verb's position. Each sentence additionally contains at least one agreement "attractor" located between the subject and the verb (the number of attractors per sample varies between 1 and 4), and all attractors are nouns of opposite number from the subject, which makes this dataset well-suited for XAI evaluation, as the evidence for the correct verb number shall be concentrated on the subject. We noticed that the original dataset from Goldberg (2019) contained 46 invalid samples, where the singular and plural verb forms were identical, which we discarded from our benchmark.

In the following we describe how we identify the subject of each sentence (i.e. the linguistic evidence we use as the ground truth for the XAI evaluation), as well as the preprocessing steps we undertook to take into account each language model's specific tokenization.

Generic ground truth. In a first step we generate the model-agnostic ground truth data. For that purpose we use Spacy's dependency parser from the english pipeline `en_core_web_trf` to identify the subject of a given verb in a sentence. We retain only samples with the syntactic dependency relation `nsubj` (i.e., "nominal subject"), thereby we aim to remove potential ambiguous cases (this step discards 1005 samples). Further, we retain only samples whose verb was identified by Spacy's part-of-speech tagger to be either of type `VBZ` ("verb, 3rd person singular present") or `VBP` ("verb, non-3rd person singular present") (thereby discarding 148 samples where the verb was not recognized as being conjugated in present tense). This leaves us with a dataset of size 28,786, from which 67% of the samples contain a "plural" verb form as the correct prediction (note that such "plural" verb forms also include some rare samples where the pronouns "I" and "you" are the subject, and thus strictly speaking would be singular cases), and 33% of the samples contain a "singular" verb form, hence the verb's number is imbalanced.

Tokenized ground truth. For each considered language model, we generate in a second step a model-specific benchmark made of tokenized stimulis and their corresponding tokenized ground truths, by taking into account each model's particular tokenizer. More precisely, we discard samples for which the verb's singular or plural inflection gets tokenized into more than one token, since the SVA prediction is based on comparing the logit scores for these two verb forms. Further, we verify that the ground truth is always shorter (in terms of number of tokens) than the input text stimuli to avoid any trivial cases for XAI evaluation. For causal language models (i.e. GPT2-like), we also discard samples where a portion of the ground truth lies *after* the verb in the sentence. Finally, we ensure that the *effective* input text (i.e. when excluding some special tokens, such as `[CLS], [SEP]` and `[MASK]` for BERT) is always longer than one token, again to avoid any trivial cases for XAI evaluation.

With the above considerations, we finally obtain for BERT a benchmark made of 28472 samples, whose input length (in terms of number of tokens) varies between 9 and 170, with mean 30, std 12, and median of 28, while the ground truth's length varies between 1 and 7 with 96.7% of the samples having a ground truth length of 1 (and 2.6% of samples a ground truth length of 2).

For GPT-2 we likewise obtain a benchmark made of 28,602 samples, whose input length (in terms of number of tokens) varies between 2 and 60, with mean 11, std 7, and median of 8, while the ground truth's length varies between 1 and 7 with 85.1% of the samples having a ground truth length of 1 (and 12.4% of samples a ground truth length of 2).

For Llama-3 we finally obtain a benchmark made of 28,629 samples, with an input length (in number of tokens) varying between 3 and 60, with mean 11, std 7, and median of 9, while the ground truth's length varies between 1 and 5 with 89.4% of the samples having a ground truth length of 1 (and 8.8% of samples a ground truth length of 2).

C Language Models

Table 3 summarizes the prediction accuracies of each model on our tokenized subject-verb agreement benchmark datasets, as well as provides various informations about the models' sizes and tokenizers.

D Proofs on implementing LRP/AttnLRP via a *modified* Gradient×Input strategy

D.1 LRP-ϵ rule for linear layers

Given a linear of the form $z_j = \sum_i z_i w_{ij} + b_j$ in the forward pass, and given the relevances of the output neurons R_j, the input neurons' relevances R_i are computed using the following LRP-ϵ rule:

$$R_i = \sum_j \frac{z_i \cdot w_{ij}}{z_j + \epsilon \cdot \text{sign}(z_j)} \cdot R_j \tag{1}$$

The term ϵ is typically a small positive numerical stabilizer. But for simplifying the derivation let's assume $\epsilon = 0$, and so: $R_i = \sum_j \frac{z_i \cdot w_{ij}}{z_j} \cdot R_j$.

Now let's assume the relevances at the layer output and input are computed via Gradient×Input, in other words it holds:

$$R_j = dz_j \cdot z_j \tag{2}$$
$$R_i = dz_i \cdot z_i \tag{3}$$

Using elementary rules of differentiation and the chain rule it holds:

$$dz_i = \sum_j dz_j \cdot w_{ij} \tag{4}$$

By incorporating Eq. 4 into Eq. 3, we obtain:

$$R_i = z_i \cdot \sum_j dz_j \cdot w_{ij} \tag{5}$$

And the replacing dz_j by its value from Eq. 2, we finally get:

$$R_i = z_i \cdot \sum_j \frac{R_j}{z_j} \cdot w_{ij} \tag{6}$$

And by rearranging terms:

$$R_i = \sum_j \frac{z_i \cdot w_{ij}}{z_j} \cdot R_j \quad \square \tag{7}$$

Hence we have shown that using Gradient×Input one can implement the LRP-ϵ rule with $\epsilon = 0$. Using the Gradient×Input strategy presents even an advantage over an explicit implementation of the LRP-ϵ rule. Indeed with Gradient×Input no fraction is involved in the computation, and hence no denominator needs to be stabilized, while with explicit LRP one has to use a non-zero ϵ stabilizer, which might introduces some noise or dampen the explanation process, as the ϵ value is kind of arbitrary, and its impact will be higher the lower the magnitude of the denominator's value.

D.2 LRP-identity rule for element-wise activation layers

Given an element-wise activation layer of the form: $z_j = g(z_i)$, with g being the activation function. The LRP-identity rule redistributes the relevance identically from the layer's output to the layer's input, thus $R_i = R_j$.

Now let's define a modified forward function for the layer of the form:

$$\hat{z}_j = z_i \cdot [\tfrac{g(z_i)}{z_i}]_{\text{detach()}} \tag{8}$$

Obviously it holds that $\hat{z}_j = z_j$, so the forward pass outcome remains unchanged.

Using elementary rules of differentiation and the chain rule it holds:

$$dz_i = [\tfrac{g(z_i)}{z_i}] \cdot d\hat{z}_j \tag{9}$$

Now assuming we compute the relevances at the layer's output as well as at the layer's input with Gradient×Input using the modified layer, we get:

$$R_i = dz_i \cdot z_i = [\tfrac{g(z_i)}{z_i}] \cdot d\hat{z}_j \cdot z_i \tag{10}$$
$$= g(z_i) \cdot d\hat{z}_j = \hat{z}_j \cdot d\hat{z}_j \tag{11}$$
$$= R_j \quad \square \tag{12}$$

Hence we have shown the LRP-identity rule can be implemented implicitly by using the *modified* Gradient×Input strategy.

Table 3: Prediction accuracy on subject-verb agreement, and model information.

Model	prediction accuracy	# params	# layers	# heads	hidden size	vocab size	tokenizer
`bert-base-uncased`	0.969	110M	12	12	768	30522	WordPiece
`bert-large-uncased`	0.974	340M	24	16	1024	same	same
`GPT2-small`	0.919	124M	12	12	768	50257	BPE
`GPT2-XL`	0.941	1.5B	48	25	1600	same	same
`Llama-3.2-1B`	0.954	1B	16	32	2048	128256	tiktoken BPE
`Llama-3.2-3B`	0.956	3B	28	24	3072	same	same

Moreover, note that one does not even need a numerical stabilizer to handle a zero-valued input in the activation layer. Indeed most considered element-wise activation functions (such as GELU or SiLU) have a zero-valued output when their input is zero. Thus one possibility to deal with a zero-valued input is to set the output manually to zero for $\hat{z}_j$ in this particular case (i.e., to a constant), hence the resulting gradient will be zero too. And as a consequence, the relevance using the Gradient×Input strategy will be zero. This is still meaningful for LRP as in such a case the output's relevance will be zero anyway, so there no relevance to redistribute backward (indeed LRP relevances are generally proportional to neurons' contributions in the forward pass, and for a subsequent linear layer a zero-valued input does not contribute to the output, hence receiving no relevance).

D.3 LRP-signal-take-all rule for product layers

Given a product layer of the following form: $z_j = z_g \cdot z_s$, where z_g is a gate neuron and z_s is a signal neuron (in the MHA attention layer the former will be the attention weight, and the latter a component of the value vector).

The LRP-signal-take-all rule redistributes all the relevance to the signal neuron, i.e. $R_s = R_j$ and $R_g = 0$.

And let's define the following modified product layer:

$$\hat{z}_j = [z_g]_{\text{detach()}} \cdot z_s \tag{13}$$

Obviously $\hat{z}_j = z_j$.

Now assuming we compute the relevances at the layer's output and input via Gradient×Input, thus it holds:

$$R_j = d\hat{z}_j \cdot \hat{z}_j \tag{14}$$

$$R_s = dz_s \cdot z_s \tag{15}$$

$$R_g = dz_g \cdot z_g \tag{16}$$

Per definition of elementary rules of differentiation and the chain rule, we have:

$$dz_s = d\hat{z}_j \cdot z_g \tag{17}$$

$$dz_g = 0 \tag{18}$$

By incorporating Eq. 17&18 into Eq. 15&16, we finally get:

$$R_s = d\hat{z}_j \cdot z_g \cdot z_s = d\hat{z}_j \cdot \hat{z}_j \tag{19}$$

$$R_g = 0 \tag{20}$$

Und by using Eq. 14:

$$R_s = R_j \tag{21}$$

$$R_g = 0 \quad \square \tag{22}$$

Hence we have shown the LRP-signal-take-all rule can be implemented implicitly by using the *modified* Gradient×Input strategy.

D.4 LRP rule for normalization layers

We illustrate this rule using the Pytorch LayerNorm layer, which is defined by:

$$z_j = \frac{z_i - E[z_i]}{\sqrt{Var[z_i] + \epsilon}} \cdot \gamma + \beta \tag{23}$$

where the parameters of the layers ϵ, γ and β are constants.

In order to extend LRP to Transformers Ali et al. (2022) propose to treat the standard deviation of the Layernorm as a constant, which can be achieved by modifying the layer in the following way:

$$\hat{z}_j = \frac{z_i - E[z_i]}{[\sqrt{Var[z_i] + \epsilon}]_{\text{detach()}}} \cdot \gamma + \beta \tag{24}$$

Obviously $\hat{z}_j = z_j$.

Further the modified layer is now a linear layer (since all operations in the layer such as the mean operation are now linear). Hence the layer can be treated similarly to Section D.1. $\square$

So overall we have shown that the LRP rule proposed for normalization layers in Transformers can be implenmented with Gradient×Input.

D.5 AttnLRP rule for softmax layers

Let us introduce some new notations to match closely the ones from Achtibat et al. (2024). So far we have mainly dealt with single neurons, now we deal with vectors. So let the input vector be $\mathbf{x}$ and the output vector be $\mathbf{s}$, and both can be indexed either by i or j.

Per defintion of the softmax operation we have:

$$s_j(\mathbf{x}) = \frac{e^{x_j}}{\sum_i e^{x_i}} \tag{25}$$

Using elementary rules of differentiation one can show that:

$$\frac{\partial s_j}{\partial x_i} = \begin{cases} s_j(1-s_j) & \text{for } i = j \\ -s_j s_i & \text{for } i \neq j \end{cases} \tag{26}$$

Now assuming the input's and output's relevances are computed via Gradient×Input, i.e.:

$$R_{s_j} = ds_j \cdot s_j \tag{27}$$
$$R_{x_i} = dx_i \cdot x_i \tag{28}$$

Using the chain rule it holds that:

$$R_{x_i} = dx_i \cdot x_i \tag{29}$$
$$= \sum_j \frac{\partial s_j}{\partial x_i} \cdot ds_j \cdot x_i \tag{30}$$

By incorporating Eq. 31 into Eq. 30, one obtains:

$$R_{x_i} = \sum_j \begin{cases} s_j(1-s_j) \cdot ds_j \cdot x_i & \text{for } i = j \\ -s_j s_i \cdot ds_j \cdot x_i & \text{for } i \neq j \end{cases} \tag{31}$$

By identifying the term from Eq. 27:

$$R_{x_i} = \sum_j \begin{cases} (x_i - s_j \cdot x_i) \cdot R_{s_j} & \text{for } i = j \\ -s_i \cdot x_i \cdot R_{s_j} & \text{for } i \neq j \end{cases} \tag{32}$$

Hence we finally arrived at the LRP rule proposed for the softmax layer in Achtibat et al. (2024). $\square$

Thus, in summary, we have provided a complete set of proofs that all LRP, resp. AttnLRP, rules used in Transformers can be implemented via a Gradient×Input approach, by simply modifying adequately parts of the non-linear layers (namely product, normalization and element-wise activation layers) and keeping all linear layers unmodified.

`ALTI-Logit:`

<|begin_of_text|> the edited versions of the g reek literary canon that we know

<|begin_of_text|> you should find out if your school participates in the program before you choose

<|begin_of_text|> these are extremely dangerous , completely unsub stant iated allegations and , as a professional journalist myself , i find

<|begin_of_text|> changing to neutral - per the recent clean up and the comments rag arding not ability below , i see

<|begin_of_text|> wiki standards from what i read

`AttnLRP:`

<|begin_of_text|> the edited versions of the g reek literary canon that we know

<|begin_of_text|> you should find out if your school participates in the program before you choose

<|begin_of_text|> these are extremely dangerous , completely unsub stant iated allegations and , as a professional journalist myself , i find

<|begin_of_text|> changing to neutral - per the recent clean up and the comments rag arding not ability below , i see

<|begin_of_text|> wiki standards from what i read

Figure 2: Exemplary heatmaps for the ALTI-Logit and AttnLRP attribution methods, we highlight the verb in green, positive relevance is mapped to red, negative to blue.