RelP: Faithful and Efficient Circuit Discovery in Language Models via Relevance Patching

Farnoush Rezaei Jafari[1,2*] **Oliver Eberle**[1,2] **Ashkan Khakzar**

Neel Nanda

[1]Machine Learning Group, Technische Universität Berlin
[2]BIFOLD – Berlin Institute for the Foundations of Learning and Data

Abstract

Activation patching is a standard method in mechanistic interpretability for localizing the components of a model responsible for specific behaviors, but it is computationally expensive to apply at scale. Attribution patching offers a faster, gradient-based approximation, yet suffers from noise and reduced reliability in deep, highly non-linear networks. In this work, we introduce *Relevance Patching* (RelP), which replaces the local gradients in attribution patching with propagation coefficients derived from Layer-wise Relevance Propagation (LRP). LRP propagates the network's output backward through the layers, redistributing relevance to lower-level components according to local propagation rules that ensure properties such as relevance conservation or improved signal-to-noise ratio. Like attribution patching, RelP requires only two forward passes and one backward pass, maintaining computational efficiency while improving faithfulness. We validate RelP across a range of models and tasks, showing that it more accurately approximates activation patching than standard attribution patching, particularly when analyzing residual stream and MLP outputs in the Indirect Object Identification (IOI) task. For instance, for MLP outputs in GPT-2 Large, attribution patching achieves a Pearson correlation of 0.006, whereas RelP reaches 0.956, highlighting the improvement offered by RelP. Additionally, we compare the faithfulness of sparse feature circuits identified by RelP and Integrated Gradients (IG), showing that RelP achieves comparable faithfulness without the extra computational cost associated with IG. Code is available at https://github.com/FarnoushRJ/RelP.git.

1 Introduction

Recent advances in machine learning continue to rely on transformer-based language models, which achieve remarkable performance across a wide range of tasks. Given their widespread adoption, understanding the internal mechanisms of these models is an important challenge for improving our ability to interpret, trust, and control them.

To address this challenge, the fields of eXplainable Artificial Intelligence (XAI) [1, 2] and, more recently, mechanistic interpretability [3, 4] have emerged, aiming to reveal the decision-making processes of such complex architectures. Early efforts in XAI focused on developing feature attribution methods, often visualized as heatmaps, to highlight relevant features in classification models. As state-of-the-art models have grown more complex, approaches that extend beyond input explanations have become increasingly important for uncovering their internal strategies. This has motivated

*Correspondence to: rezaeijafari@campus.tu-berlin.de

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: ResponsibleFM.

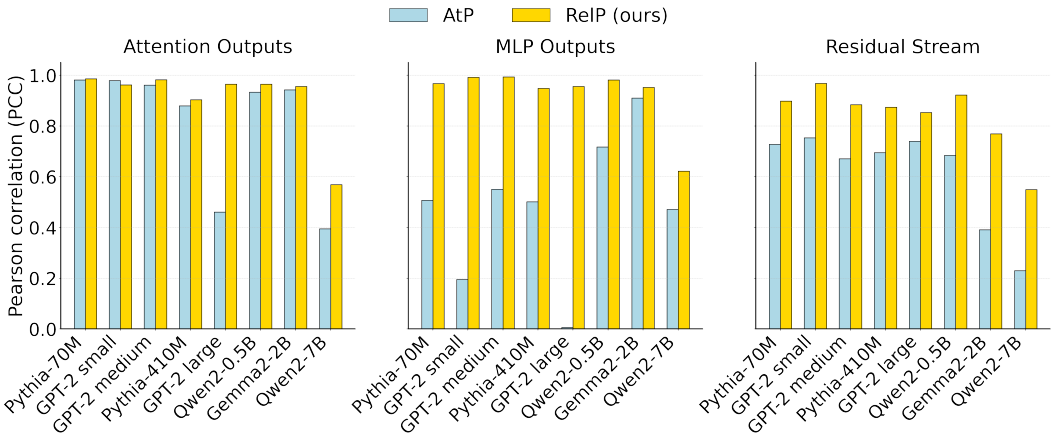

Figure 1: Pearson correlation coefficient (PCC) between activation patching and attribution patching (AtP) or relevance patching (RelP), computed over 100 IOI prompts for three GPT-2 model sizes (Small, Medium, Large), two Pythia models (70M, 410M), two Qwen2 models (0.5B, 7B), and Gemma2-2B. A higher value of PCC represents higher alignment with activation patching results.

a shift toward exploring the complex internal structure of neural networks through methods such as concept-based interpretability [5, 6], higher-order explanations [7, 8, 9], and circuit discovery [3, 10]. Herein, a central goal of interpretability research is the reliable localization of specific model components responsible for particular functions, algorithms or behaviors [3, 4, 11, 12, 13]. To identify components that are causally involved in inference, mechanistic interpretability has proposed activation patching, also known as causal mediation analysis [14, 15, 16]. Activation patching replaces the activations of selected components in an original input with those from a patch input, allowing direct causal testing of the contribution of a component to the model's output.

As these patching-based interventions are computationally expensive, it is challenging to scale them, especially for larger models. To improve efficiency, attribution patching [17] was introduced as a fast gradient-based approximation to activation patching. Using gradients to estimate the influence of each component on the model's output, attribution patching significantly reduces the computational cost compared to traditional perturbation-based approaches. However, this efficiency comes at the expense of robustness and accuracy, particularly in deep models like state-of-the-art LLMs [18]. This issue is not unique to attribution patching. Many gradient-based attribution methods [19, 20, 21, 22, 23] suffer from a lack of robustness in large networks, often due to noisy and unreliable gradients [24] that undermine the quality of explanations [25, 26, 27].

Alternative attribution methods have been proposed to overcome these limitations. One of the most widely used methods is Layer-wise Relevance Propagation (LRP) [28], developed within the theoretical framework of Deep Taylor Decomposition (DTD) [29]. LRP works by propagating the model's output backward through the network, redistributing the relevance of each component to its inputs according to layer-specific propagation rules. These rules are designed to enforce desirable properties such as conservation of total relevance, sparsity of explanations, and robustness to noise.

Building on these insights, we propose *Relevance Patching* (RelP), a technique that replaces local gradients in attribution patching with propagation coefficients achieved using LRP. RelP enables more faithful localization of influential components in large models, without sacrificing scalability. Similar to attribution patching, RelP requires two forward passes and one backward pass. We demonstrate the effectiveness of RelP in the context of the Indirect Object Identification task, showing that it consistently outperforms attribution patching across a range of architectures and model sizes, including GPT-2 {Small, Medium, Large} [30], Pythia-{70M, 410M} [31], Qwen2-{0.5B, 7B} [32], and Gemma2-2B [33], with the strongest gains on the residual stream and MLP outputs. In addition, we apply RelP to recover sparse feature circuits responsible for Subject–Verb Agreement in the Pythia-70M model. Compared to Integrated Gradients [21], RelP demonstrates comparable faithfulness in identifying meaningful circuits while also offering greater computational efficiency.

2 Related Works

Activation and Attribution Patching Techniques Activation patching is a causal mediation technique that tests the role of model components by replacing activations from an original run with those from a patch run [34, 35]. It has been widely used in interpretability studies [15, 36, 37, 38], with variants such as causal tracing, which perturbs activations with Gaussian noise [16], and path patching, which extends interventions to multi-step computation paths [39, 40]. To improve efficiency, attribution patching (AtP) uses gradient-based attribution instead of full interventions, requiring only two forward passes and one backward pass [17], and AtP* further enhances reliability while maintaining scalability [18]. Building on this line of work, we propose Relevance Patching (RelP), which also targets efficiency but replaces the local gradients in attribution patching with propagation coefficients computed using Layer-wise Relevance Propagation (LRP) [28], improving faithfulness to activation patching while maintaining the computational advantages of AtP.

Circuit Analysis Circuit analysis methods aim to uncover subnetworks that drive particular model behaviors. Activation patching, used in ACDC [41], identifies critical edges in the computation graph but is computationally expensive. It has been scaled to large models [42] and adapted to study the effects of fine-tuning [43]. Sparse Autoencoders (SAEs) offer an alternative by learning disentangled, interpretable latent features [44, 45]. Methods such IFR [46] and EAP [47] instead use gradient-based attributions to perform circuit discovery in a more efficient way. Finally, attribution-guided pruning with LRP has been shown to support both circuit discovery and model compression [48]. Our proposed Relevance Patching (RelP) also builds on LRP but stays within the patching framework, providing faithful and efficient attribution that approximates activation patching.

Gradient-Based and Propagation-Based Attribution Methods Local gradients have long been used to explain the behavior of non-linear models [23]. Through backpropagation, they enable the efficient computation of saliency maps [28, 49], which have been extensively studied in the context of vision models. Saliency maps derived from raw gradients [49] capture the sensitivity of model predictions to small perturbations in input features. When these gradients are scaled by the corresponding input values, they yield Gradient$\times$Input [22] explanations. Grad-CAM [20] aggregates class-specific gradients over the final convolutional feature maps, yielding coarse but discriminative heatmaps. Since raw gradients are often noisy, SmoothGrad [19] averages saliency maps over random perturbations, PathwayGrad [50] finds a sub-network (pathway) critical for the output and propagates gradients through that pathway, and Integrated Gradients [21] integrates along a baseline-to-input path to mitigate saturation and enforce key attribution axioms. Beyond gradient-based methods, propagation-based approaches such as LRP [28] and DeepLIFT [51] redistribute the model's output backward through the network using local propagation rules, which can offer more faithful attribution. For a comprehensive review of explanation methods and attribution techniques, see [1, 2, 52, 53].

3 Background

From Activation to Attribution Patching Let $\mathcal{M} : X \to \mathbb{R}^V$ be a decoder-only transformer model that maps an input sequence $x \in X := \{1, ..., V\}^T$ to a vector of output logits over a vocabulary of size V. We represent the model as a directed computation graph $G = (N, E)$, where each node $n \in N$ corresponds to a distinct model component, and each edge $(n_1, n_2) \in E$ represents a direct computational dependency. The activation of a component n when processing input x is denoted $n(x)$.

Let D be a distribution over prompt pairs $(x_{\text{original}}, x_{\text{patch}})$, where x_{original} is a representative task sample that captures the behavior of interest, and x_{patch} serves as a reference input used to introduce counterfactual perturbations for causal intervention. A metric $\mathcal{L} : \mathbb{R}^V \to \mathbb{R}$ quantifies the output behavior of the model.

The causal contribution $c(n)$ of a component $n \in N$ is defined as the expected effect of replacing its activation in the original input with the corresponding activation from the patch input, i.e.,

$$c(n) := \mathbb{E}_{(x_{\text{original}}, x_{\text{patch}}) \sim D} \left[\mathcal{L}\left(\mathcal{M}(x_{\text{original}} \mid \text{do}(n \leftarrow n(x_{\text{patch}})))\right) - \mathcal{L}\left(\mathcal{M}(x_{\text{original}})\right) \right]. \tag{1}$$

This definition follows the causal intervention framework, where $\text{do}(n \leftarrow n(x_{\text{patch}}))$ denotes overwriting the activation of n with its value under the patch input. Evaluating $c(n)$ directly for all $n \in N$ is computationally prohibitive for large-scale models, motivating the development of efficient approximation methods such as attribution patching (AtP) [17]:

$$\hat{c}_{\text{AtP}}(n) := \mathbb{E}_{(x_{\text{original}}, x_{\text{patch}}) \sim D} \left[\left(n(x_{\text{patch}}) - n(x_{\text{original}}) \right)^{\top} \left. \frac{\partial \mathcal{L}(\mathcal{M}(x_{\text{original}}))}{\partial n} \right|_{n = n(x_{\text{original}})} \right]. \quad (2)$$

AtP (Eq. 2) approximates the effect of replacing component n's activation from the original input with that from the patch input, without explicitly running the patched model, by leveraging gradients.

Layer-wise Relevance Propagation (LRP) Layer-wise Relevance Propagation (LRP) [28] is a framework for interpreting neural network predictions. The central idea is to assign a relevance score $\mathcal{R}$ to each component or input feature, quantifying its contribution to the model's output or to any metric defined from the output. These relevance scores are then propagated backward through the network according to layer-specific rules. The propagation follows the principle of relevance conservation [28, 29], ensuring that the total relevance at each layer is completely redistributed to the preceding layer, ultimately attributing the chosen metric to the input features.

To formalize this process, let $a_i^{(l-1)}$ with $i \in \{1, ..., n\}$ denote the activations in layer $l-1$. These activations serve as inputs to a layer represented by $f^{(l)} : \mathbb{R}^n \to \mathbb{R}^m$, resulting in outputs $a_j^{(l)}$, $j \in \{1, ..., m\}$. A local first-order Taylor expansion around a reference point $\tilde{a}^{(l-1)}$ expresses each output as

$$f_j^{(l)}(a^{(l-1)}) \approx \sum_i J_{ji}^{(l)} \, a_i^{(l-1)} + \tilde{b}_j^{(l)}, \qquad J_{ji}^{(l)} = \left. \frac{\partial f_j^{(l)}}{\partial a_i^{(l-1)}} \right|_{\tilde{a}^{(l-1)}}.$$

Here, $J_{ji}^{(l)}$ is the local Jacobian, describing how activations $a_i^{(l-1)}$ influences outputs $a_j^{(l)}$, while $\tilde{b}_j^{(l)}$ collects constant terms and approximation errors.

This decomposition motivates the LRP redistribution step. Each component j in layer l is assigned a relevance score $\mathcal{R}_j^{(l)}$, which is redistributed to its inputs in proportion to their contributions to the activation of component j. Here, this redistribution from upper-layer activations to component i is expressed through a **propagation coefficient** ρ_i, which aggregates the contributions from all connected upper-layer components j. The relevance of component i in layer $l-1$ is then given by

$$\mathcal{R}_i^{(l-1)} = a_i^{(l-1)} \, \rho_i.$$

The propagation coefficient ρ_i depends on the local Jacobian $J^{(l)}$, the activations $a^{(l-1)}$, and the relevance scores at layer l (i.e., $\mathcal{R}^{(l)}$). Intuitively, ρ_i acts as a filter that determines which parts of the input activations are considered relevant.

The layer-wise redistribution of relevance can be repeated until the input layer is reached or stopped at an intermediate layer of interest. The procedure begins with an initial relevance signal $\mathcal{R}_c^{(L)}$, typically defined using a metric computed from the model outputs, commonly the logit of the true, predicted, or any target class c at the output layer [28, 29]. Contrastive approaches have also been proposed, which explain the difference between class logits to highlight features that strongly influence changes in the model's prediction [26, 54].

4 RelP: Relevance Patching for Mechanistic Analysis

Gradient-based attribution methods have been extensively studied in the XAI literature [19, 20, 21, 22, 23]. Despite their simplicity and model-agnostic applicability, these methods often face challenges when applied to large neural networks due to noisy and less reliable gradients [25, 27]. LRP mitigates these shortcomings by propagating the network's output $\mathcal{M}(x)$, or any metric $\mathcal{L}$ defined from the output, backward through the network in a structured and theoretically grounded manner. Depending on the layers in a given model architecture, different propagation rules have been proposed. These rules are typically derived from gradient analyses of model components [26, 29, 55], with specific

Propagation Rules for Transformers

Layer	Propagation Rule	Implementation Trick
LayerNorm / RMSNorm	LN-rule [26]	$y_i = \frac{x_i - E[x]}{[\sqrt{\epsilon + \mathrm{Var}[x]}]_{\text{const.}}}$
GELU / SiLU	Identity-rule [55]	$x \odot [\Phi(x)]_{\text{const.}}$
Linear	0-rule, ϵ-rule, or γ-rule [52]	–
Attention	AH-rule [26]	$y_j = \sum_i x_i [A_{ij}]_{\text{const.}}$
Multiplicative Gate	Half-rule [55, 57]	$0.5 \cdot (x \odot g(x)) + 0.5 \cdot [(x \odot g(x))]_{\text{const.}}$

Table 1: A summary of propagation rules for Transformer layers. Components marked with $[]_{\text{const.}}$ are treated as constants, typically using `.detach()` in PyTorch. In this table, x and y denote the input and output of a layer, respectively, and A represents the attention weights.

choices guided by common desiderata for explanations, such as sparsity, reduced noise, and relevance conservation [52, 56].

Specialized propagation rules have been developed to handle diverse nonlinear components (e.g., activation functions, attention mechanisms, normalization layers) and architectural nuances across deep network families [7, 8, 27, 52, 55, 57, 58]. An overview of relevant rules for transformer models is shown in Table 1.

As attribution patching relies on gradients to approximate the effect of substituting hidden activations ("patching in"), it is also susceptible to the noise and approximation errors inherent to gradient-based attribution methods, particularly in very large models [24, 29]. To address this, we introduce *Relevance Patching* (RelP).

Formally, RelP follows the structure of standard attribution patching (AtP). In AtP, the contribution of a component is computed as the dot product between the change in its activation, caused by replacing the original input with a patch input, and the gradient of the metric $\mathcal{L}$ with respect to that component, evaluated at the original input. RelP modifies this procedure by substituting the gradient term with the LRP-derived propagation coefficient (introduced in Section 3) for that component. The resulting contribution score is defined as

$$\begin{aligned}\hat{c}_{\text{RelP}}(n) &:= \mathbb{E}_{(x_{\text{original}}, x_{\text{patch}}) \sim D}\left[\left(n(x_{\text{patch}}) - n(x_{\text{original}})\right)^\top \rho(\mathcal{L}(\mathcal{M}(x_{\text{original}})))\Big|_{n(x_{\text{original}})} \right] \\ &= \mathbb{E}_{(x_{\text{original}}, x_{\text{patch}}) \sim D}\left[\mathcal{R}_{\text{RelP}}(\mathcal{L}(\mathcal{M}(x_{\text{original}})))\Big|_{n(x_{\text{original}})} \right],\end{aligned} \tag{3}$$

where $\rho(\mathcal{L}(\mathcal{M}(x_{\text{original}})))$ and $\mathcal{R}_{\text{RelP}}(\mathcal{L}(\mathcal{M}(x_{\text{original}})))$ denote, respectively, the propagation coefficient and the relevance score for component n.

RelP preserves the efficiency of attribution patching while improving faithfulness. Unlike Marks et al. [44], which relies on Integrated Gradients [21] for more accurate approximations at higher computational cost, RelP provides scalable and faithful localization of influential components.

Propagation Rules Table 1 summarizes the propagation rules applicable to core components of transformer architectures. As discussed in Section 3, relevance conservation is a key property in LRP, ensuring that the sum of relevance scores is preserved across layers. This prevents relevance from being lost or artificially introduced during propagation. Many of the specialized rules in Table 1 are designed to address situations where this property could fail, and are often derived from gradient analyses of individual model components.

In attention heads, attention outputs depend on inputs directly and via attention weights A_{ij}, which are also input-dependent. Correlations between these terms can break conservation and the AH-rule preserves it by treating A_{ij} as constants, effectively linearizing the heads [26]. LayerNorm's centering and variance-based scaling can cause "relevance collapse", which the LN-rule mitigates this by treating $(\sqrt{\epsilon + \mathrm{Var}[x]})^{-1}$ as constant, allowing the operation to be seen as linear and preserving

IOI Templates

Template #	Sentence Template
1	Then, [B] and [A] went to the [PLACE]. [B] gave a [OBJECT] to [A].
2	When, [B] and [A] went to the [PLACE]. [B] gave a [OBJECT] to [A].
3	After [B] and [A] went to the [PLACE], [B] gave a [OBJECT] to [A].

Table 2: Templates used for generating the IOI samples.

conservation [26]. For multiplicative gates, the Half-rule splits relevance equally to avoid spurious doubling [55, 57]. Activation functions (e.g., GELU, SiLU) can disrupt conservation since their output is a nonlinear transformation of the input, the Identity-rule addresses this by treating the non-linear component as constant [55]. For linear layers, the 0-rule is equivalent to Gradient $\times$Input [22]. The ϵ-rule and γ-rule extend the 0-rule, the ϵ-rule is applied when sparsity and noise reduction are desired, while the γ-rule is used when it is preferable to emphasize positive contributions over negative ones [52]. For additional rules specifically tailored to transformers, we refer the readers to [26, 27, 59].

5 Experiments

We evaluate the effectiveness of RelP by benchmarking it against attribution patching (AtP), on the Indirect Object Identification (IOI) task, across three GPT-2 variants (Small, Medium, and Large), two Pythia models (70M and 410M), two Qwen2 models (0.5B and 7B), and Gemma2-2B. Furthermore, we assess RelP's ability to recover sparse feature circuits underlying Subject–Verb Agreement in the Pythia-70M model, comparing its performance against Integrated Gradients (IG) as a more accurate gradient-based baseline. Additional details of our experimental setup are provided in Appendix A.

5.1 Evaluating RelP on the IOI Task

The objective of this experiment is to evaluate how well attribution patching (AtP) and relevance patching (RelP) approximate the ground-truth effects captured by activation patching (AP) on the IOI task. This task involves determining which entity in a sentence functions as the indirect object, typically the recipient or beneficiary of the direct object.

To enable controlled evaluations, we construct 100 prompt pairs. Each pair consists of an original prompt (with the correct indirect object) and a patch variant in which the subject and indirect object are swapped, keeping all other tokens the same. All prompts are generated from structured templates [10] (Table 2) to ensure consistency across examples. We define the metric $\mathcal{L}$ as the difference in logits between the original and patch targets. The methods RelP, AtP, and AP are each applied to the residual stream, attention outputs, and MLP outputs. Since the resulting attribution scores are computed across hidden dimensions, we aggregate them by summing over these dimensions. To quantify how well AtP and RelP approximate AP, we compute the Pearson correlation coefficient (PCC) between their results and those of AP.

The results of this experiment are presented in Figure 1. As shown, RelP consistently outperforms AtP in a range of architectures and model sizes, including GPT-2 {Small, Medium, Large}, Pythia-{70M, 410M}, Qwen2-{0.5B, 7B}, and Gemma2-2B. The performance gap is especially pronounced in the MLP output and residual stream analyses. Detailed numerical results are listed in Table 5 in Appendix B.

Qualitative differences between AtP and RelP are also visible in Figure 2. Nanda [17] suggests that attribution patching fails notably in the residual stream, primarily due to its large activations and the nonlinearity introduced by LayerNorm, which significantly disrupts the underlying linear approximation. It also performs poorly on MLP0, since this layer in GPT-2 Small functions as an "extended embedding", and Nanda [17] has shown that linear approximations in this layer may become unstable, leading to misleading insights. As can be seen in Figure 2, the RelP results align

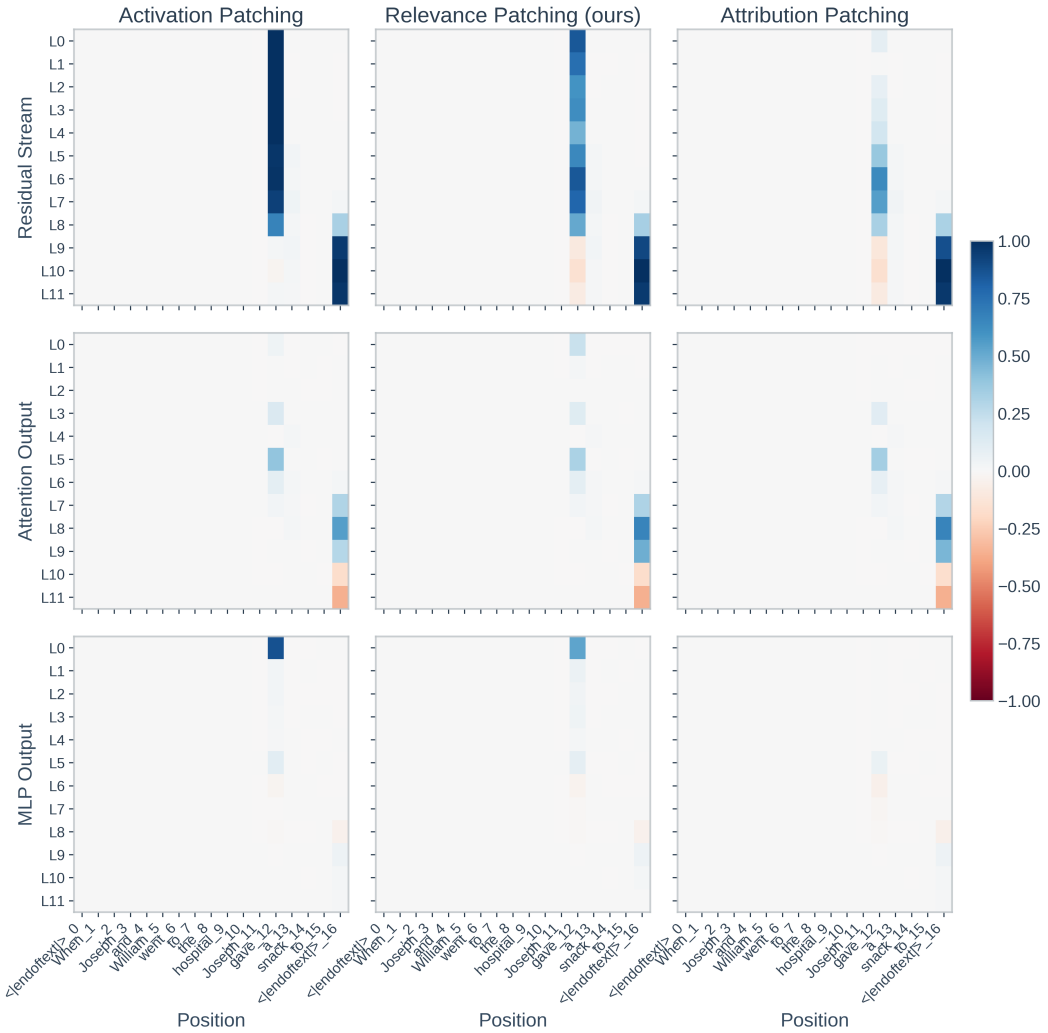

Figure 2: Qualitative comparison showing how accurately relevance patching (RelP) and attribution patching (AtP) approximate the effects of activation patching in GPT-2 Small. RelP shows notably better alignment in the residual stream and at MLP0, where AtP's estimates are less reliable.

more closely with those from activation patching, especially when considering the residual stream and MLP outputs (e.g., MLP0). Further qualitative results are provided in Appendix C.

Overall, we find that the proposed method, RelP, consistently achieves strong correlations with activation patching and is often clearly superior to standard attribution patching. This underscores the effectiveness of the tailored propagation rules in the LRP framework, which provide a reliable and computationally efficient approximation of the computationally expensive activation patching method across models and components commonly studied in mechanistic interpretability research.

5.2 Sparse Feature Circuits for Subject-Verb Agreement

Sparse feature circuits are small, interconnected groups of interpretable features that jointly drive specific behaviors in language models. Analyzing these circuits allows us to understand how models combine meaningful components to solve tasks, offering insight into the underlying mechanisms behind their decisions. In this experiment, we aim to uncover feature circuits involved in the subject-verb agreement task, a linguistic evaluation that tests a model's ability to correctly match a verb's inflection (e.g., singular or plural) to the grammatical number of its subject.

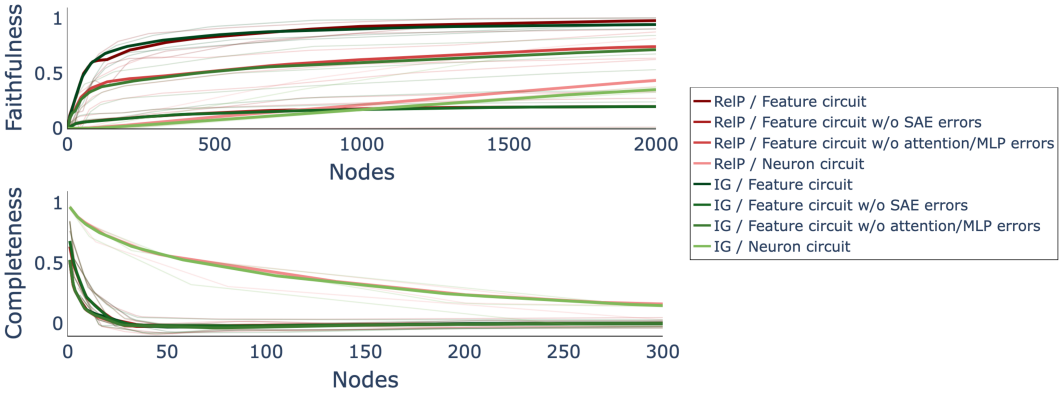

Figure 3: Faithfulness and completeness scores for circuits, evaluated on held-out data. Faint lines show individual circuits for structures from Table 3, while the bold lines indicate the average across all structures. An ideal circuit has a faithfulness score of 1 and a completeness score of 0. While Integrated Gradients (IG) requires multiple integration steps (steps=10 in this experiment), RelP achieves comparable faithfulness scores without any additional computational cost.

Subject-Verb Agreement Templates

Template Type	Example x_{original}	Example Output Metric
Within Relative Clause (RC)	The athlete that the managers	$p(\text{likes}) - p(\text{like})$
Across Relative Clause (RC)	The athlete that the managers like	$p(\text{do}) - p(\text{does})$
Across Prepositional Phrase (PP)	The secretaries near the cars	$p(\text{has}) - p(\text{have})$

Table 3: These templates involve contrastive pairs of inputs that differ only in the grammatical number of the subject, with the model's task being to choose the appropriate verb inflection.

To identify and analyze these circuits, we use features discovered using sparse autoencoders (SAEs), which provide fine-grained, human-interpretable units. We follow [44] and use their SAEs trained on Pythia-70M language model and their sparse features in our experiments. These sparse features serve as the fundamental units for circuit analysis. To assess their influence on the model's output, Marks et al. [44] employed efficient linear approximations of activation patching, such as Integrated Gradients (IG), to estimate the contributions of individual features and their interactions. Although IG provides greater faithfulness than attribution patching, it requires multiple forward and backward passes, resulting in higher computational cost. Our experiments show that RelP achieves faithfulness on par with attribution patching, without incurring additional computational overhead.

We use three templatic datasets (Table 3), where tokens at the same positions serve similar roles, allowing us to average node and edge effects across examples while retaining positional information. Consistent with [44], we assess the identified sparse feature circuits using **faithfulness** and **completeness** evaluation metrics [10]. Additionally, we compare our sparse feature circuits with neuron circuits, constructed by applying the same discovery method directly to individual neurons rather than to SAE features, serving as a baseline for evaluation.

Faithfulness measures how much of the model's original performance is captured by the discovered circuit, relative to a baseline where the entire model is mean-ablated. It is calculated as $\frac{\mathcal{L}(C)-\mathcal{L}(\emptyset)}{\mathcal{L}(M)-\mathcal{L}(\emptyset)}$, where $\mathcal{L}(C)$ is the metric when only the circuit C is active, $\mathcal{L}(\emptyset)$ is the metric of the fully mean-ablated model, and $\mathcal{L}(\mathcal{M})$ is the metric of the full model.

Completeness evaluates how much of the model's behavior is not explained by the discovered circuit. It is defined as the faithfulness of the circuit's complement, $(\mathcal{M} \backslash C)$. In other words, if the complement of the circuit still explains a lot of the behavior, then the original circuit is not complete in capturing the mechanism.

We report faithfulness scores for feature and neuron circuits as the node threshold T_N is varied (Figure 3). The threshold keeps only nodes with contribution scores above T_N, retaining the most relevant ones. Consistent with [44], small feature circuits explain most of the model's behavior: in Pythia-70M, about 100 features account for the majority of performance, whereas roughly 1,500 neurons are needed to explain half.

SAE error nodes summarize reconstruction errors in the SAE, making them fundamentally different from single neurons. To analyze their impact, we evaluate faithfulness after removing all SAE error nodes, as well as those originating from attention and MLP modules. Consistent with findings by Marks et al. [44], we observe that removing residual stream SAE error nodes severely disrupts model performance, whereas removing MLP and attention error nodes has a less pronounced effect.

Notably, the circuits identified by our RelP method achieve faithfulness scores comparable to those obtained with IG, while offering greater computational efficiency. In our experiments, IG requires 10 integration steps, increasing computational overhead. In contrast, RelP matches the efficiency of AtP, requiring only two forward passes and one backward pass, while achieving faithfulness scores on par with IG. In the case of neuron circuits and feature circuits without attention/MLP errors, we observe an improvement in the faithfulness of RelP compared to IG. This combination of faithfulness and efficiency makes RelP a more effective and practical approach for uncovering sparse feature circuits in large language models.

Following [44], we also evaluate completeness (Figure 3) and observe that ablating only a few nodes from feature circuits can drastically reduce model performance, whereas hundreds of neurons are needed for a similar effect for Pythia. This highlights the efficiency of the identified sparse feature circuits in capturing critical model behavior, as well as the usefulness of RelP in locating them.

6 Conclusion

In this paper, we introduced Relevance Patching (RelP) as an enhancement over the standard attribution patching method. RelP preserves the overall algorithmic structure of attribution patching while replacing its local gradient signal with the propagation coefficient derived from Layer-wise Relevance Propagation (LRP). This modification enables RelP to more accurately approximate the causal effects captured by activation patching, while remaining computationally efficient.

In our experiments on the Indirect Object Identification task, we demonstrated that RelP exhibits stronger alignment with activation patching across residual stream, attention, and MLP outputs compared to attribution patching. Furthermore, in comparing sparse feature circuits identified by RelP and Integrated Gradients (IG), we showed that RelP achieves comparable faithfulness without incurring the additional computational cost associated with IG. Our study thus demonstrates that methods developed for feature attribution can be effectively integrated into mechanistic interpretability, helping to advance our understanding of modern foundation models.

7 Limitations and Future Work

Applying RelP to localize relevant model components requires choosing appropriate rules within the LRP framework, which introduces some model-specific overhead compared to model-agnostic attribution methods. In our experiments, RelP consistently outperformed attribution patching on residual stream and MLP outputs, while gains for attention outputs were smaller. This suggests that a deeper analysis of gradient structure in self-attention, combined with more careful rule selection, could improve attribution faithfulness. Prior works have examined these challenges in the context of input-level feature attribution [26, 27, 55], but systematic studies of internal layers remain limited. For our experiments, we used small- and medium-scale open-access models, which allow full access to internals and gradients. The tasks were chosen to match standard circuit analysis benchmarks, relying on predefined original–patch input pairs and single-token prediction metrics. As a result, circuit discovery was limited to moderately complex behaviors. Extending RelP to more challenging settings, such as free-form text generation or scenarios without known counterfactual inputs, remains an important direction for future work.

Acknowledgments and Disclosure of Funding

We acknowledge support by the Federal Ministry of Research, Technology and Space (BMFTR) for BIFOLD (ref. 01IS18037A). F.RJ. was partly supported by MATS 8.0 program during which foundational experiments for this work were completed.

References

[1] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021. doi: 10.1109/JPROC.2021.3060483.

[2] Luca Longo, Mario Brcic, Federico Cabitza, Jaesik Choi, Roberto Confalonieri, Javier Del Ser, Riccardo Guidotti, Yoichi Hayashi, Francisco Herrera, Andreas Holzinger, et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301, 2024.

[3] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

[4] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

[5] Pattarawat Chormai, Jan Herrmann, Klaus-Robert Müller, and Grégoire Montavon. Disentangled explanations of neural network predictions by finding relevant subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11):7283–7299, 2024.

[6] Laura Kopf, Nils Feldhus, Kirill Bykov, Philine Lou Bommer, Anna Hedström, Marina MC Höhne, and Oliver Eberle. Capturing polysemanticity with PRISM: A multi-concept feature description framework. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, and Grégoire Montavon. Building and interpreting deep similarity models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1149–1161, 2020.

[8] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7581–7596, 2022.

[9] Thomas Schnake, Farnoush Rezaei Jafari, Jonas Lederer, Ping Xiong, Shinichi Nakajima, Stefan Gugler, Grégoire Montavon, and Klaus-Robert Müller. Towards symbolic xai — explanation through human understandable logical relationships between features. *Inf. Fusion*, 118(C), April 2025. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102923.

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

[11] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in neural information processing systems*, 36:27223–27250, 2023.

[12] Takuya Ito, Murray Campbell, Lior Horesh, Tim Klinger, and Parikshit Ram. Quantifying artificial intelligence through algorithmic generalization. *Nature Machine Intelligence*, pages 1–11, 2025.

[13] Oliver Eberle, Thomas Austin McGee, Hamza Giaffar, Taylor Whittington Webb, and Ida Momennejad. Position: We need an algorithmic understanding of generative AI. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

[14] Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

[15] Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, 2022.

[16] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

[17] Neel Nanda. Attribution patching: Activation patching at industrial scale. 2022. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

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

[19] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, 2017.

[21] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[22] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning (ICML)*, 2017.

[23] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[24] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning (ICML)*, volume 70, pages 342–350. PMLR, 2017.

[25] Ann-Kathrin Dombrowski, Christopher J. Anders, Klaus-Robert Müller, and Pan Kessel. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022. ISSN 0031-3203.

[26] Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. XAI for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning (ICML)*, 2022.

[27] Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *International Conference on Machine Learning (ICML)*, 2024.

[28] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[29] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

[30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[31] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[32] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[33] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[34] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[35] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, 2020.

[36] Paul Soulos, R. Thomas McCoy, Tal Linzen, and Paul Smolensky. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020.

[37] Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020.

[38] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[39] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023.

[40] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*, 2019.

[41] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023.

[42] Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

[43] Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations (ICLR)*, 2024.

[44] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations (ICLR)*, 2025.

[45] Dmitrii Kharlapenko, Stepan Shabalin, Arthur Conmy, and Neel Nanda. Scaling sparse feature circuits for studying in-context learning. In *International Conference on Machine Learning (ICML)*, 2025.

[46] Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[47] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023.

[48] Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Reduan Achtibat, Patrick Kahardipraja, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. Attribution-guided pruning for compression, circuit discovery, and targeted correction in llms. *arXiv preprint arXiv:2506.13727*, 2025.

[49] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014.

[50] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Neural response interpretation through the lens of critical pathways. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13528–13538, 2021.

[51] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, 2017.

[52] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019.

[53] Ashkan Khakzar, Pedram Khorsandi, Rozhin Nobahari, and Nassir Navab. Do explanations explain? model knows best. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10244–10253, 2022.

[54] Oliver Eberle, Ilias Chalkidis, Laura Cabello, and Stephanie Brandl. Rather a nurse than a physician - contrastive explanations under investigation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[55] Farnoush Rezaei Jafari, Grégoire Montavon, Klaus-Robert Müller, and Oliver Eberle. MambaLRP: Explaining selective state space sequence models. In *Advances in neural information processing systems*, 2024.

[56] William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. ISBN 978-3-642-77927-5.

[57] Leila Arras, José Arjona-Medina, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter, and Wojciech Samek. Explaining and interpreting LSTMs. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 211–238, 2019.

[58] Simon Letzgus, Patrick Wagner, Jonas Lederer, Wojciech Samek, Klaus-Robert Müller, and Grégoire Montavon. Toward explainable artificial intelligence for regression models: A methodological perspective. *IEEE Signal Processing Magazine*, 39(4):40–58, 2022.

[59] Yarden Bakish, Itamar Zimerman, Hila Chefer, and Lior Wolf. Revisiting lrp: Positional attribution as the missing ingredient for transformer explainability. *arXiv preprint arXiv:2506.02138*, 2025.

A Experimental Details

A.1 LRP Implementation

We applied the LN-rule [26] for LayerNorm/RMSNorm, the Identity-rule [55] for nonlinear activation functions, and the LRP-0 rule [52] for linear transformations. The Half-rule [55, 57] was applied only to the Qwen2 model family and Gemma2-2B, as other models used in our experiments did not include multiplicative gating operations. No attention-specific rule (e.g., AH-rule) was used. Full details are given in Table 4.

Propagation Rule	GPT-2 Family	Pythia Family	Qwen2 Family	Gemma2-2B
LN-rule [26]	✓	✓	✓	✓
Identity-rule [55]	✓	✓	✓	✓
0-rule [52]	✓	✓	✓	✓
Half-rule [55, 57]	×	×	✓	✓
AH-rule [26]	×	×	×	×

Table 4: Propagation rules applied in our LRP implementation across different model families. A green tick (✓) indicates that the rule was applied, while a red cross (×) indicates that it was not used.

A.2 Further Details on Subject-Verb Agreement Experiment

In this experiment, we used 300 samples, structured according to Table 3, for circuit discovery, and 100 held-out samples for evaluating faithfulness and completeness. Consistent with [44], the first one-third of each circuit was excluded from evaluation, since components in early model layers are typically responsible for processing specific tokens, which may not consistently appear across training and test splits.

B Quantitative results

The exact numerical values corresponding to the IOI experiment, visualized in Figure 1, are reported in Table 5.

	Residual Stream		Attention Outputs		MLP Outputs	
Model	AtP	RelP (ours)	AtP	RelP (ours)	AtP	RelP (ours)
GPT-2 small	0.753	**0.968**	**0.979**	0.962	0.195	**0.992**
GPT-2 medium	0.671	**0.884**	0.961	**0.982**	0.551	**0.993**
GPT-2 large	0.740	**0.853**	0.461	**0.965**	0.006	**0.956**
Pythia-70M	0.728	**0.898**	0.981	**0.986**	0.507	**0.967**
Pythia-410M	0.695	**0.874**	0.879	**0.903**	0.501	**0.948**
Qwen2-0.5B	0.684	**0.922**	0.933	**0.965**	0.717	**0.981**
Qwen2-7B	0.230	**0.549**	0.395	**0.569**	0.471	**0.622**
Gemma2-2B	0.391	**0.769**	0.942	**0.956**	0.910	**0.952**

Table 5: Pearson correlation coefficient (PCC) between activation patching and attribution patching (AtP) or relevance patching (RelP), computed over 100 IOI prompts for three GPT-2 model sizes (Small, Medium, Large), two Pythia models (70M, 410M), two Qwen2 models (0.5B, 7B), and Gemma2-2B. A higher value of PCC represents higher alignment with activation patching results.

C Further Qualitative Results

In Section 5, we presented the qualitative differences between AtP and RelP for the GPT-2 Small model. In this section, we extend our analysis by providing additional experimental results for other models. It is evident that RelP provides a more accurate approximation to activation patching, particularly for the residual stream and MLP outputs.

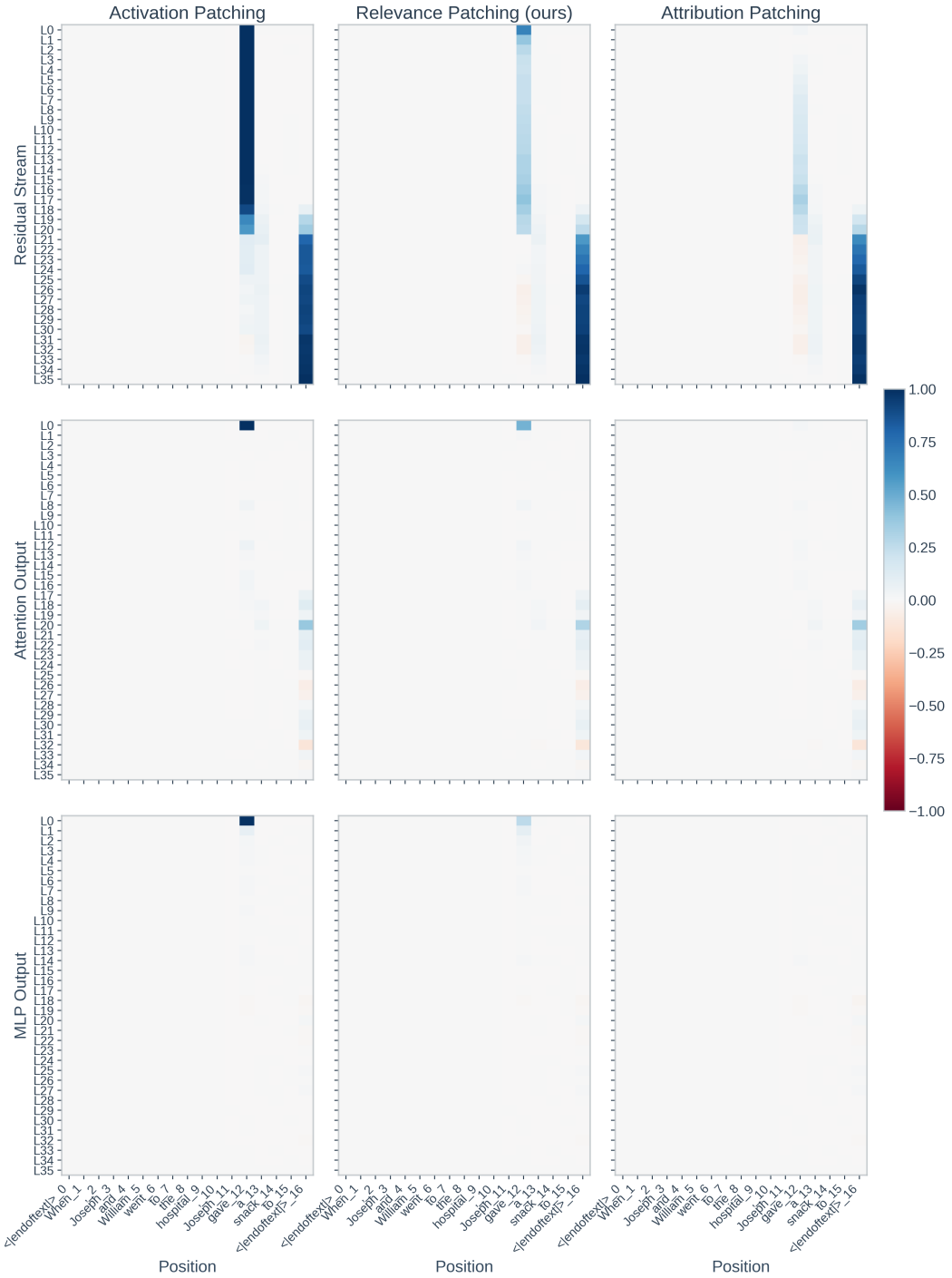

Figure 4: Qualitative comparison showing how accurately relevance patching (RelP) and attribution patching (AtP) approximate the effects of activation patching in GPT-2 Large. RelP shows notably better alignment in the residual stream and at MLP0, where AtP's estimates are less reliable.

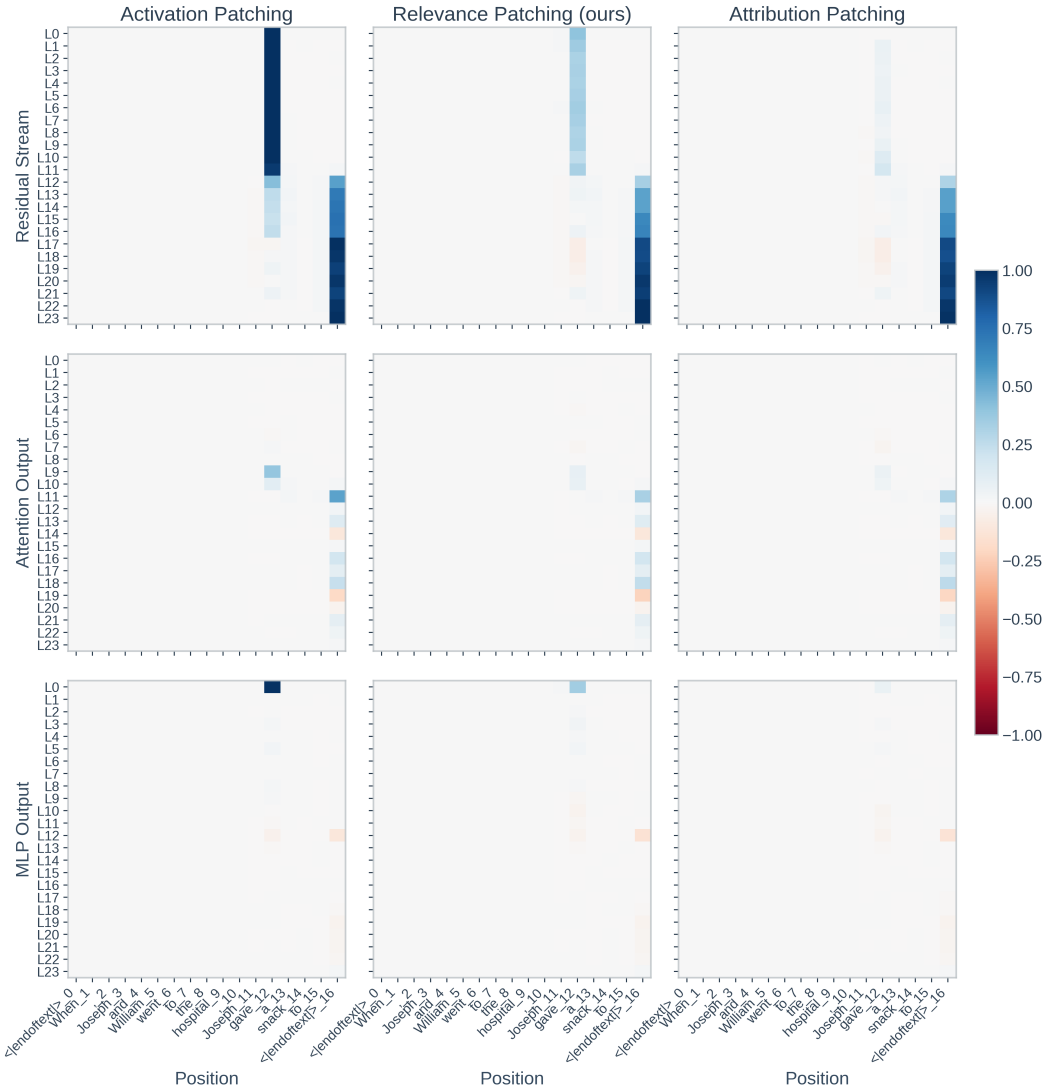

Figure 5: Qualitative comparison showing how accurately relevance patching (RelP) and attribution patching (AtP) approximate the effects of activation patching in Pythia-410M. RelP shows notably better alignment in the residual stream and at MLP0, where AtP's estimates are less reliable.