CAN LAYER-WISE RELEVANCE PROPAGATION CAPTURE COMPETITION AMONG TRANSFORMER LOGITS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Layer-wise Relevance Propagation (LRP) is a widely used framework for attributing Transformer predictions to input tokens. However, existing Transformer-based LRP methods typically propagate relevance from a single output logit, which implicitly assumes that the prediction is governed by that logit in isolation. This assumption is limiting, since what determines the prediction is not the absolute magnitude of any individual logit but the relative relationships among logits. Intuitively, a factor that contributes equally and positively to all logits does not affect the final decision and therefore should not be attributed as important, a property that current algorithms generally struggle to model. We formalize this limitation from two complementary perspectives in transformers: low-coupling projection heads and highly coupled parameters in deeper layers, showing that single-logit relevance cannot faithfully reflect inter-class competition. To address this, we propose a top-k differential weighting strategy that constructs scalar targets by contrasting the largest logit with its strongest competitors, thereby explicitly incorporating competition into the relevance propagation process. Extensive perturbation experiments show that our method, by down-weighting common-mode evidence that contributes equally to competing logits, helps LRP capture the truly target-specific features that primarily influence the target logit, and consistently outperforms state-of-the-art baselines. Our code is available at https://anonymous.4open.science/r/RI-LRP-6AF0.

1 INTRODUCTION

Deep learning models have achieved remarkable performance on a wide range of complex tasks, but their internal decision-making mechanisms often remain opaque. This lack of transparency severely impedes the deployment of such models in domains with stringent traceability requirements, such as healthcare Mishra (2026); Chhetri et al. (2025) and finance Tuan & Sanyal (2026). Consequently, the development of effective explanation methods Rashid et al. (2026b); Wu et al. (2026); Li & Ting (2026) has become a focal point for both academia and industry.

Among modern architectures, Transformers Vaswani et al. (2017) have established a dominant position in natural language processing and computer vision. Their parameter scales often reach billions, imposing additional pressure on interpretability research. Researchers Atad et al. (2026); Ahmed et al. (2026); Mersha & Kalita (2026b) have proposed various attribution strategies customized to Transformer architectures. Among these, Layer-wise Relevance Propagation (LRP) Bach et al. (2015) has emerged as one of the most promising approaches Agbo Eje et al. (2026). It propagates relevance signals backward through the network, redistributing importance across neurons according to predefined rules, and has since seen widespread application across various tasks Hu et al. (2026); Peng et al. (2025); Huang et al. (2026). Compared with gradient-based methods Simonyan et al. (2013), which often suffer from gradient vanishing or numerical instability, LRP demonstrates superior robustness and precision in tracking decision information flow.

Despite these advances, our analysis reveals a fundamental limitation in existing LRP-based attribution methods for Transformers. The model's final prediction hinges on the relative competition among logits, but standard LRP isolates only the maximum logit for propagation, overlooking how input features may simultaneously influence both the predicted class and its rivals. This narrow focus can produce misleading attribution signs. A feature that suppresses a competitor more strongly than the predicted class actually bolsters the decision margin and should be positively credited, but standard

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

LRP may mark it as negative since it attends solely to the absolute change of the target logit. We formalize this shortcoming from two perspectives. At the linear classification head, backpropagating the predicted logit alone cannot perceive any information along competing directions. At the deeper layers, the heuristic redistribution rules employed by LRP deviate from the exact training gradients, so relevance follows a different path from the one that training constrains.

To remedy this, we propose a scheme for LRP-based Transformer explainability that explicitly models relative influence among competing logits. We construct a scalar target by contrasting the maximum logit against a uniformly weighted sum of its top competitors, introducing competitive information directly into the attribution objective. The predicted class logit receives full positive weight, the top competitors receive negative uniform weights so that their contributions are subtracted in the differential target, and all others are zeroed out. Leveraging the linearity of LRP with respect to the scalar target, our attribution decomposes as a weighted sum of individual logit attributions. This competitive formulation naturally down-weights features that equally promote the predicted class and its competitors, as they lack discriminative utility. The perturbation benchmarks show improved AUAC over the evaluated baselines, and the competitive window is selected on a small in-distribution validation subset. We additionally test whether this ranking advantage transfers to answer-span localization.

In summary, this work makes three main contributions:

- We give a projection-head counterexample showing that unary-logit relevance retains common-mode, non-discriminative evidence, and distinguish this issue from deep-layer behavior.
- To alleviate the limitation of unary relevance propagation, we propose a novel LRP scheme that explicitly models relative influence among competing logits.
- Extensive experiments in the NLP and vision domains show that our method consistently surpasses state-of-the-art LRP approaches, validating that explicit competition modeling is crucial.

2 Related Work

Research on interpretability for deep neural networks can be broadly categorized into perturbation-based, local surrogate-based, and backpropagation-based methods.

2.1 Perturbation and Local Surrogate-based Methods

Perturbation-based methods occlude or replace input features to observe output changes Zeiler & Fergus (2014); Fong & Vedaldi (2017). SHAP Lundberg & Lee (2017) introduces Shapley values into feature attribution. However, these methods Rashid et al. (2026a); Guwal et al. (2025) require multiple forward passes with the cost growing linearly with the dimensionality of the feature. Local surrogate-based methods such as LIME Ribeiro et al. (2016) and LORE Guidotti et al. (2018) fit interpretable linear models locally. Although model-agnostic, they Li et al. (2025a); Angiulli et al. (2025) fail to capture latent representations of original model.

2.2 Backpropagation-based Attribution Methods

Backpropagation-based methods propagate output signals backward through the computational graph, offering computational efficiency. Input$\times$Gradient Simonyan et al. (2013) is among the earliest but suffers from gradient shattering Balduzzi et al. (2017). SmoothGrad Smilkov et al. (2017) and Integrated Gradients Sundararajan et al. (2017) mitigate this via multiple sampling, but show limited effectiveness on large-scale Transformers.

LRP Bach et al. (2015) formulates attribution as layer-by-layer relevance redistribution from output to input. Montavon et al. (2017) established Deep Taylor Decomposition as its theoretical foundation, where each layer's redistribution corresponds to a first-order Taylor expansion. A rich family of propagation rules has since emerged: the z^+-rule Montavon et al. (2015) for ReLU networks, the γ-rule Samek et al. (2022) for convolutional layers, and the $\alpha\beta$-rule Bach et al. (2015) controlling positive and negative contributions. RI-LRP leaves the propagation rule untouched and modifies

only the output seed: by contrasting the winning logit with competing class logits, it down-weights common-mode evidence that a unary seed would wrongly credit. This differs from RAP Nam et al. (2020), whose within-layer redistribution acts at a different stage, and from WB-LRP Li et al. (2025b), whose weight-dependent baselines reshape the rule itself.

2.3 LRP Extensions for Transformer Architectures

Extending LRP to Transformers presents dual challenges from bilinear layers and softmax nonlinearities in attention. Ali et al. (2022) blocked gradient flow through softmax, but this bypasses rather than resolves propagation issues. Other works Chefer et al. (2021); Mersha & Kalita (2026a) combine gradients with LRP-style metrics for attention aggregation, though they yield only approximate attributions and neglect countervailing evidence.

AttnLRP Achtibat et al. (2024) first achieves high-fidelity attribution for full Transformers, although subsequent studies reveal theoretical deficiencies. You et al. (2025) show its bilinear rule violates implementation invariance, while Bakish et al. (2026) redefine the input space as position-token pairs with sound propagation rules for positional encodings. More recently, Jantsch et al. (2026) proposed dual path attribution for SwiGLU-Transformers, and Suzuki et al. (2026) introduced layer-wise integrated gradients composing contributions within layer. For state-space models, Rezaei Jafari et al. (2024) propose MambaLRP to identify components responsible for unfaithful explanations.

3 Preliminary

Let a deep neural network be implemented as the mapping:

$$f : \mathbb{R}^d \to \mathbb{R}^C, \quad \mathbf{x} \mapsto \mathbf{z} = (z_1, z_2, \ldots, z_C),$$

where d and C represent the dimensions of the input and output spaces, respectively. $\mathbf{z}$ denotes the output logit vector, which undergoes a softmax function to obtain the final probability distribution. The predicted target is $c^*(\mathbf{x}) = \arg\max z_c(\mathbf{x})$ and the maximum logit value is denoted as $z_{c^*}(\mathbf{x})$. The attribution task aims to assign relevance scores to input features in order to explain the model's decision. Within the LRP framework, this task is equivalent to constructing an explanatory function:

$$\mathcal{R} : \mathbb{R}^d \to \mathbb{R}^d, \quad \mathbf{x} \mapsto \mathbf{R}(\mathbf{x}),$$

such that $\mathbf{R}(\mathbf{x}) = (R_1(\mathbf{x}), \ldots, R_d(\mathbf{x}))$ aims at two criteria: (i) conservation for exactly conserving rules, where $\sum_i R_i(\mathbf{x})$ equals the interpreted output target; and (ii) faithfulness, where larger-magnitude relevance identifies features that matter to the chosen target. Stabilizers and postprocessing can break exact conservation.

The existing LRP method of transformer restricts the attribution target to the maximum logit for the current sample. Consequently, the relevance vector for the input features is given by

$$\mathbf{R}^{(\text{LRP})}(\mathbf{x}) = \Phi_{\text{LRP}}\left(\mathbf{x},\ z_{c^*}(\mathbf{x})\right),$$

where Φ_{LRP} denotes the LRP backpropagation operator, which propagates the output scalar $z_{c^*}(\mathbf{x})$ back to the input layer by layer according to its local redistribution rules.

4 Analyses of Existing Methods

The model's final output is a probability distribution, and the prediction decision is determined by the relative relationships among logits rather than the absolute magnitude of any single logit. Standard LRP explains only the composition of z_{c^*}, yet fails to capture the relative variations among logits, which we term **Relative Influence**. A given input feature can simultaneously suppress both the target logit z_{c^*} and a competing logit z_k. If the suppression of z_k is more pronounced, then from a relative perspective the feature actually promotes the softmax margin of c^* over k and should receive a positive contribution. Standard LRP may incorrectly assign a negative relevance because it only cares about the absolute size change of the current logit, violating the semantic meaning of LRP score signs: positive contributions should support the current decision, while negative contributions should oppose it. Equivalently, a factor contributing equally and positively to all competing logits does

not affect the relative decision among them, and should therefore not be credited as target-specific evidence; the unary target provides no mechanism to enforce this.

We distinguish projection-head parameters from deep-layer parameters. The projection-head analysis gives a direct counterexample to unary targets, and the deep-layer analysis shows that training gradients and LRP relevance follow different paths.

4.1 Insufficiency of Projection Head

Consider only the projection head of the last layer. Let the hidden representation be $\mathbf{h}$, with logits

$$z_c = \mathbf{h}^\top \mathbf{W}_c,$$

where $\mathbf{W}_c$ is the weight vector for class c. For a target class y and a competing class k, standard LRP propagation of the target score z_y, under approximate linear propagation Achtibat et al. (2024), yields

$$\mathbf{R}_y(\mathbf{x}) \propto \mathbf{h} \odot \mathbf{W}_y,$$

where $\odot$ denotes the Hadamard product. This attribution propagates solely along $\mathbf{W}_y$, completely disregarding $\mathbf{W}_k$: since the output weight vectors are structurally independent, propagating the isolated scalar z_y captures no information along $\mathbf{W}_k$. Consequently, the unary target cannot distinguish a direction that changes only the absolute level of z_y from one that changes the relative margin $z_y - z_k$; both are treated as evidence about the target.

We now exhibit a perturbation that leaves the binary decision unchanged yet alters the unary attribution. Add the same vector $\mathbf{u}$ to every class weight, i.e. $\mathbf{W}_c \mapsto \mathbf{W}_c + \mathbf{u}$ for $c \in \{y, k\}$. In the two-class case the softmax depends only on the logit difference,

$$p_y = \sigma(z_y - z_k),$$

and both logits shift by the same amount, $z'_c = z_c + \mathbf{h}^\top \mathbf{u}$, so the difference, the probability, and the decision boundary are exactly preserved:

$$z'_y - z'_k = z_y - z_k, \qquad p'_y = p_y.$$

However, the unary LRP attribution of the target logit becomes

$$\mathbf{R}_{y'}(\mathbf{x}) \sim \mathbf{h} \odot (\mathbf{W}_y + \mathbf{u}) = \mathbf{R}_y(\mathbf{x}) + \mathbf{h} \odot \mathbf{u}.$$

Standard LRP is thus sensitive to a perturbation carrying no discriminative information: it credits common-mode evidence, which contributes equally to the competing logits and is therefore irrelevant to the relative decision, as if it were target-specific evidence.

At the final layer, the relative influence among competing classes is thus entirely absent from the relevance decomposition. Appendix A extends this argument to the multi-class setting under an all-class common perturbation.

4.2 Insufficiency of Deep-layer Parameters

Extending to deep parameters, the cross-entropy loss explicitly couples all class logits during training:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x},y)} \left[-\log \frac{\exp(z_y(\mathbf{x};\theta))}{\sum_{c=1}^{C} \exp(z_c(\mathbf{x};\theta))} \right],$$

where $z_y(\mathbf{x};\theta)$ denotes the logit under parameter θ. Its gradient with respect to any parameter θ_i is

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \mathbb{E}_{(\mathbf{x},y)} \left[\sum_{c=1}^{C} (p_c(\mathbf{x}) - \mathbf{1}_{c=y}) \frac{\partial z_c(\mathbf{x};\theta)}{\partial \theta_i} \right],$$

where $p_c(\mathbf{x}) = \mathrm{softmax}_c(\mathbf{z}(\mathbf{x};\theta))$. This gradient weights each logit's derivative by the probability residual $p_c - \mathbf{1}_{c=y}$.

In contrast, LRP propagates relevance from the target scalar z_{c^*} using hand-designed local rules, e.g.:

$$R_i^{(l)} = \sum_j \frac{\rho(a_i \cdot w_{ij})}{\sum_{i'} \rho(a_{i'} \cdot w_{i'j}) + \epsilon} R_j^{(l+1)},$$

where $R_i^{(l)}$ is the relevance of neuron i in layer l, a_i its activation, w_{ij} the weight from neuron i in layer l to neuron j in layer $l+1$, $\rho(\cdot)$ predefined rule functions Achtibat et al. (2024), and ϵ a small stabilizer. These rules are not derived from the actual gradient of z_{c^*} with respect to intermediate activations, but are designed for conservation and interpretability. Consequently, training gradients and LRP relevance follow fundamentally different paths: the former adheres to exact chain-rule derivatives, while the latter relies on a surrogate redistribution scheme.

Training constrains the aggregated derivatives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \mathbb{E}\left[\sum_c (p_c - \mathbf{1}_{c=y}) \frac{\partial z_c}{\partial W_{ij}^{(l)}}\right],$$

whereas LRP uses local redistribution ratios such as $\rho(a_i W_{ij}) / \sum_{i'} \rho(a_{i'} W_{i'j})$. These quantities are generally different, so LRP relevance need not match training sensitivity; the discrepancy is structural. Training gradients constrain only the aggregated, probability-weighted sum of logit derivatives, which is invariant under any reparameterization preserving the input-output function, whereas LRP ratios are defined on raw weights and activations. The model also admits a flat parameter subspace in which every logit is preserved for every input, yet LRP relevance is not constant because the stabilized ϵ-rule is not scale-equivariant (Appendix B). This weakens implementation invariance and indicates that the unary seed only partially reflects training sensitivity: it credits common-mode components invisible to the relative decision while missing target-specific sensitivity.

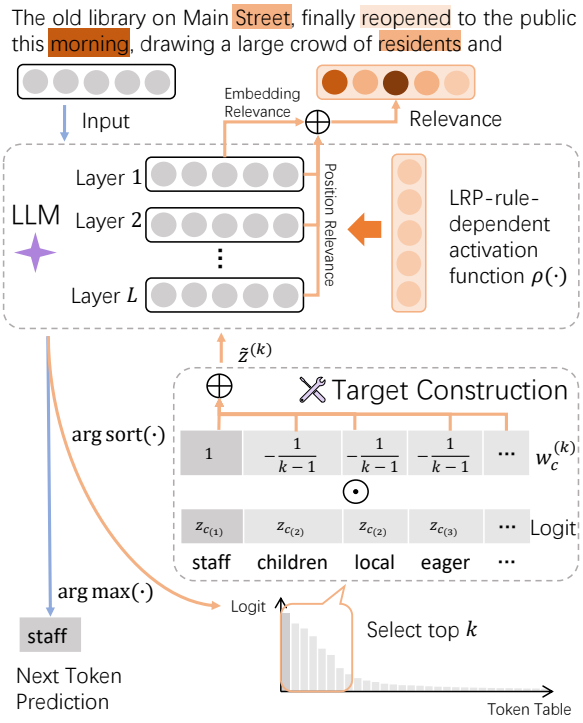

Figure 1: Visualization of our method for propagating relevance. Blue arrows indicate the forward path, while orange arrows represent the relevance propagation path.

5 Proposed Method

We propose an output-target modification for LRP-based Transformer explainability that explicitly models relative influence among competing logits, as illustrated in Fig.1 and detailed in Algorithm 1 of Appendix C.

Top-k differential target construction. We introduce competitive information by reconstructing the attribution target. Let $\mathbf{z} \in \mathbb{R}^C$ denote the output logits of a C-class classifier for input $\mathbf{x}$. Define $\text{argsort}(\mathbf{z}) = (c_{(1)}, \ldots, c_{(C)})$ such that $z_{c_{(1)}} \geq \cdots \geq z_{c_{(C)}}$, and let $\mathcal{S}_k(\mathbf{z}) = \{c_{(1)}, \ldots, c_{(k)}\}$ be the index set of the k largest logits.

The predicted logit receives weight $+1$, the remaining $k-1$ top logits each receive $-1/(k-1)$, and all others receive zero. The coefficients sum to zero,

$$1 - (k-1) \cdot \frac{1}{k-1} = 0,$$

so a common offset a added to every selected logit leaves the target exactly unchanged: $\tilde{z}^{(k)}(\mathbf{z}+a\mathbf{1}) = \tilde{z}^{(k)}(\mathbf{z})$. Common-mode evidence shared by all selected logits is thus eliminated at the output seed rather than merely attenuated, which is the seed-level counterpart of the down-weighting described in the abstract; nonlinear postprocessing may still reintroduce part of it in the final heatmap. This is the coefficient used by the reported implementation.

For $k > 1$, define $\mathbf{w}^{(k)} \in \mathbb{R}^C$:

$$w_c^{(k)} = \begin{cases} 1, & c = c_{(1)}, \\ -\dfrac{1}{k-1}, & c \in \mathcal{S}_k \setminus \{c_{(1)}\}, \\ 0, & c \notin \mathcal{S}_k. \end{cases}$$

For $k = 1$, $w_{c_{(1)}}^{(1)} = 1$ and $w_c^{(1)} = 0$ for all $c \neq c_{(1)}$. The weighted aggregation target is

$$\tilde{z}^{(k)} = \sum_{c \in \mathcal{S}_k(\mathbf{z})} w_c^{(k)} z_c = z_{c_{(1)}} - \frac{1}{k-1} \sum_{c \in \mathcal{S}_k \setminus \{c_{(1)}\}} z_c,$$

and using this differential target as the seed, LRP yields

$$\mathbf{R}_{\text{raw}}^{(k)}(\mathbf{x}) = \Phi_{\text{LRP}}\left(\mathbf{x},\ \tilde{z}^{(k)}\right).$$

For a fixed input and a redistribution rule linear in its incoming relevance, the raw signed propagation is linear in the output seed. Hence the scalar target can be implemented in a single backward pass, and before nonlinear postprocessing its attribution can be rewritten as a weighted sum of individual-logit attributions:

$$\begin{aligned} \mathbf{R}_{\text{raw}}^{(k)}(\mathbf{x}) &= \sum_{c=1}^{C} w_c^{(k)} \cdot \Phi_{\text{LRP}}(\mathbf{x}, z_c) \\ &= \Phi_{\text{LRP}}(\mathbf{x}, z_{c^*}) - \frac{1}{k-1} \sum_{i=2}^{k} \Phi_{\text{LRP}}(\mathbf{x}, z_{c_{(i)}}). \end{aligned}$$

This is how competition enters relevance propagation: the rule is unchanged, but the propagated scalar is now a differential target contrasting the winning logit with its strongest competitors. Since the coefficients sum to zero, a common offset a applied to every selected logit cancels exactly at the output seed, $\tilde{z}^{(k)}(\mathbf{z} + a\mathbf{1}) = \tilde{z}^{(k)}(\mathbf{z})$. In the linear-head counterexample, adding the same vector $\mathbf{u}$ to every selected output weight induces the same common offset $\mathbf{h}^\top \mathbf{u}$ in the selected logits and leaves the differential target unchanged. Common-mode evidence is thus eliminated at the seed level rather than attenuated by a fixed factor; nonlinear postprocessing may still reintroduce it in the final heatmap.

When $k = 1$, $\tilde{z}^{(1)} = z_{c_{(1)}}$, recovering standard single-logit LRP. For $k > 1$, the attribution subtracts $1/(k-1)$ of the summed attribution of competing logits, incorporating competing hypotheses when explaining the predicted class. A feature promoting both the maximum and the runner-up class may not discriminate between them and should receive a lower score. Uniform weighting assumes no calibration among the selected alternatives and requires no temperature. The runner-up captures the nearest pairwise margin, while lower-ranked logits broaden the contrast. We test probability- and inverse-gap-weighted competitors with equal total negative mass in a separate control, and examine sensitivity to k in Section 6.5.

This competitive formulation directly addresses the projection-head counterexample by replacing an absolute logit target with a contrastive one. For the projection head, the effective direction is $\mathbf{W}_{c^*} - (k-1)^{-1} \sum_{i=2}^{k} \mathbf{W}_{c_{(i)}}$, which incorporates the selected competing directions rather than only $\mathbf{W}_{c^*}$. A feature adding the same amount to all selected logits has zero net contribution to the differential seed before postprocessing: its contribution to the winning logit is exactly canceled by the sum of its contributions to the $k-1$ competitors. For deep layers, $\tilde{z}^{(k)}$ cancels the same selected-logit common-mode component at the seed level. In this sense the method down-weights common-mode evidence and biases the resulting attribution toward the target-specific component that distinguishes the winner from its competitors, which is the behavior claimed in the abstract.

Positional relevance accumulation. Positional features carry spatial and structural signals distinct from semantic token content. We retain the positional propagation rules of PE-LRP (Bakish et al., 2026): after redistributing neuron relevance at layer l, $\text{PosLRP}(P_{i,l}, R_i; \varepsilon)$ denotes the positional contribution for token i. We accumulate these across layers and add them to the semantic input relevance.

Table 1: Zero-shot perturbation performance of Llama-3.1-8B-Instruct. G and P denote generation and pruning AUAC, respectively; higher is better. Bold indicates the best result in each column.

Method	ARC-Easy		Wikipedia		MATH-500		CodeJudge-Eval	
	G. AUAC	P. AUAC	G. AUAC	P. AUAC	G. AUAC	P. AUAC	G. AUAC	P. AUAC
Random	.1065	.1026	.1510	.1558	.1514	.1508	.1081	.1051
Input×Grad (ICLR'14)	.2826	.2727	.4202	.4190	.4201	.4172	.1522	.1517
IG (PMLR'17)	.3213	.3116	.4342	.4329	.4865	.4835	.1398	.1395
SHAP (NeurIPS'17)	.3372	.3208	.3365	.3420	.3440	.3361	.0878	.0869
AttnRoll (ACL'20)	.3938	.3824	.4592	.4582	.5535	.5512	.2422	.2417
CP-LRP (ICML'22)	.3425	.3327	.5631	.5612	.5926	.5856	.3338	.3332
AttnLRP (ICML'24)	.3472	.3404	.5605	.5598	.5447	.5386	.4019	.4008
PE-LRP (NeurIPS'26)	.3626	.3542	.5738	.5720	.5782	.5709	.3567	.3555
DFAX (AAAI'26)	.1998	.1894	.1905	.1938	.2243	.2212	.0835	.0831
RI-LRP (ours)	**.4140**	**.4048**	**.6219**	**.6196**	**.6091**	**.6039**	**.5060**	**.5053**

The final heatmap combines semantic and positional components:

$$\mathbf{R} = (R_1, \ldots, R_d) + \mathbf{R}_{\text{pos}},$$

where $(R_1, \ldots, R_d)$ are the semantic attributions from the input layer. Absolute-value aggregation and the positive-only positional combination used in the experiments are nonlinear; the linear decomposition above applies to raw signed relevance. The weighted output seed allows one relevance-propagation pass; top-k selection and weighting add work at the output head, while the Transformer propagation cost remains the same.

Table 2: Perturbation performance on IMDB for fine-tuned language models. G and P denote generation and pruning AUAC; higher is better. Bold indicates the best value for each model and metric.

Model	Method	G. AUAC	P. AUAC
LLaMa-2 7B	Random	.5188	.5348
	CP-LRP	.5417	.5858
	AttnLRP	.5542	.5938
	PE-LRP	.5635	.5922
	RI-LRP (ours)	**.5765**	**.5967**
LLaMa-2 13B	Random	.5646	.5699
	CP-LRP	.6321	.6817
	AttnLRP	.6754	.6922
	PE-LRP	.6746	.6920
	RI-LRP (ours)	**.6871**	**.6950**
LLaMa-3 8B	Random	.5686	.5518
	CP-LRP	.6932	.7125
	AttnLRP	.6915	.7205
	PE-LRP	.7287	.7234
	RI-LRP (ours)	**.7477**	**.7285**

6 Experiments

We demonstrate the effectiveness and efficiency of the proposed method through perturbation experiments. The hyperparameter specifications for all methods are detailed in Appendix D.

6.1 Datasets

We evaluate our method on four zero-shot tasks with LLaMa-3 Grattafiori et al. (2024): **Wikipedia** for long-text next-token prediction, **ARC-Easy** for multiple-choice scientific reasoning Clark et al. (2018), **MATH-500** for mathematical problem-solving Hendrycks et al. (2021), and **CodeJudge-Eval** for code understanding Zhao et al. (2025). For fine-tuned evaluation, we use **IMDB** for binary sentiment classification Maas et al. (2011) with fine-tuned LLaMa-2 Llama (2023) and LLaMa-3 models. We also validate on **ImageNet** using DeiT Touvron et al. (2021) for vision transformers.

6.2 Baselines

We compare our method (RI-LRP) against a wide range of baselines, including traditional attribution approaches such as Input×Gradient Simonyan et al. (2013), Integrated Gradients (IG) Sundararajan et al. (2017), SHAP Lundberg & Lee (2017), and Attention Rollout (AttnRoll) Abnar & Zuidema (2020). For layer-wise relevance propagation variants, we further incorporate CP-LRP Ali et al. (2022), AttnLRP Achtibat et al. (2024), and PE-LRP Bakish et al. (2026), which extend LRP with contextual propagation rules, attention-layer applicability, and positional encoding integration,

respectively. Additionally, we include DFAX Li & Ting (2026), a recent method that leverages kernel density estimation of data distribution to interpret classifiers. A Random baseline is also incorporated as a sanity check in the quantitative evaluation.

6.3 Evaluation metrics

We adopt generation and pruning perturbation tests Bakish et al. (2026) as our evaluation metrics. Generation perturbation incrementally adds input tokens in descending order of their relevance scores, measuring the Area Under the Activation Curve (AUAC) of the model's predicted probability on the target output; an effective attribution method should yield a rapid rise in probability as high-relevance tokens are introduced. Pruning perturbation progressively masks tokens in ascending order of relevance, and a faithful method should maintain prediction stability as low-impact tokens are removed.

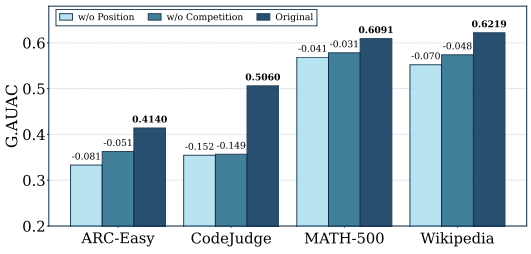

Figure 2: Contribution of competition and positional relevance, measured by perturbation tests across four datasets.

Following the work of Ali et al. (2022), AUAC is employed as the quantitative metric for both tasks. Generation and pruning traverse the same retained-token subsets in reverse order. SQuAD token IoU provide a complementary localization measure in Section F.

6.4 Main result

Table 1 presents the quantitative evaluation results of four dataset settings. RI-LRP outperforms all baseline methods, achieving the highest AUAC scores on every dataset. Against the strongest baseline in generation AUAC, RI-LRP improves by 5.1% on ARC-Easy, 8.4% on Wikipedia, 2.8% on MATH-500, and 31.2% on CodeJudge-Eval (relative improvements).

Among the baselines, LRP variants are particularly competitive on CodeJudge-Eval, whereas SHAP and DFAX are weak. Input×Gradient and Integrated Gradients provide smaller gains over Random than the strongest LRP baselines. RI-LRP achieves the highest AUAC across all language settings. Its competitive target is evaluated separately from positional relevance in the ablation study (Fig. 2).

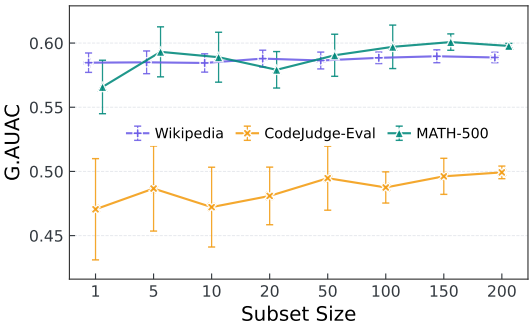

Figure 3: Test generation AUAC after selecting k on validation subsets of varying sizes, averaged over 10 runs; shading shows the standard deviation.

Evaluation on Finetuned Models We further evaluate RI-LRP on LLaMa-2 (7B and 13B) and LLaMa-3 (8B) models finetuned on the IMDB dataset.

As summarized in Table 2, these results confirm that RI-LRP scales effectively with increased capacity and remains robust in different models and finetuning setups. Random also obtains higher AUAC after fine-tuning, which reflects the more concentrated predictive signal of the fine-tuned models.

Evaluation on ImageNet. Appendix E reports results on a 50-class ImageNet-1K subset for three DeiT model sizes, providing an additional vision-domain evaluation.

We divide the dataset into validation and test sets. The optimal k is searched on validation sets of varying sizes and then evaluated on the test set. Fig. 3 reports the test generation AUAC after selecting k on validation subsets of different sizes, averaged over 10 runs with the standard deviation shown as shading. The standard deviation decreases rapidly as the validation subset grows, dropping below

0.005 on all datasets at 200 samples. The mean performance stabilizes around 50 to 100 samples, indicating that a validation set of 200 instances suffices to reliably identify a well-performing k.

6.5 Discussion of Top-k

Fig. 4 presents the sweep across k. For Wikipedia and MATH-500, performance increases steadily and peaks around $k = 1000$ before gradually decreasing, suggesting that contrasting a broad set of competing hypotheses benefits open-domain and mathematical reasoning tasks. CodeJudge-Eval achieves its optimum at a much smaller $k = 10$, with a sharp decline thereafter, indicating that code understanding tasks favor narrow competitive windows to avoid over-penalization. The eventual degradation at very large k is consistent with including increasingly weak competitors: their average is a less local proxy for the decision boundary and can dilute the contrastive signal.

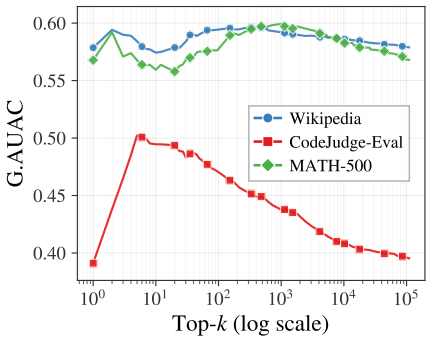

Figure 4: Test AUAC as a function of k on a log scale (80 points) for three datasets. Higher values indicate better attribution.

A 200-sample validation subset is sufficient to select a k that yields stable test performance; we do not claim that it recovers the test-optimal k in every case. When labels are unavailable, the pairwise choice $k = 2$ is a natural default for explaining the nearest decision boundary; a larger k can be selected on an unlabeled held-out set using a fixed perturbation protocol. The rank-k logit gap and cumulative top-k probability mass provide useful diagnostics for such a choice.

Appendix G provides logit-rank and tail diagnostics supporting this finite-window view: at $k = 1000$, the median winner-to-cutoff gap is 12.06 logits and the mean omitted mass is 1.86%, though the omitted tail still exceeds the winner in 2.5% of contexts. These diagnostics thus indicate that the window captures most probability mass while still leaving a small non-negligible tail.

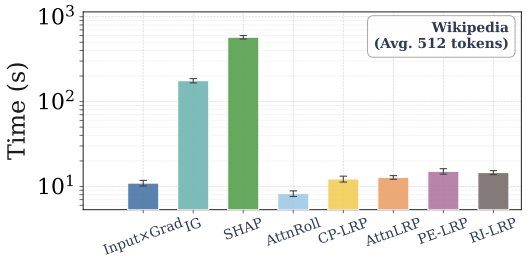

Figure 5: Runtime (log scale) on 50 Wikipedia samples, grouped by token count; error bars show the standard deviation over 10 runs. Lower values indicate faster methods, and shorter error bars indicate more stable runtime.

6.6 Computational Efficiency

The runtime results in Fig. 5 demonstrate that our method achieves a favorable efficiency-accuracy trade-off. Our method maintains runtime within a narrow margin of the LRP-based baselines, while both SHAP and IG exhibit substantially higher computational costs that scale poorly with input length. This overhead remains small relative to accuracy gains and grows only mildly as the input length increases across all datasets. A detailed comparison across multiple datasets is provided in Appendix H.

7 Conclusions

This paper reveals a shortcoming in LRP-based attribution: propagating only the maximum logit ignores relative influence among competing classes. We formally analyze this limitation and introduce RI-LRP, a contrastive framework that differentially weights top competing logits for propagation. This formulation remedies the output-seed deficiency and provides a simple target-level modification that can be combined with existing LRP rules, while deep-layer rule sensitivity remains an open issue. Experiments show RI-LRP consistently outperforms the state-of-the-art method, and for a fixed evaluation protocol the optimal window is reliably selected via a small validation set. Future work will extend RI-LRP to model debugging and adversarial robustness, leveraging competitive attribution to pinpoint misleading features.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and writing assistance. Specifically, we used these tools to improve the clarity, grammar, and readability of the manuscript, and to help draft and revise portions of the text. We have not used generative AI tools for generating research ideas, designing the proposed method, deriving theoretical results, implementing the code, conducting experiments, or producing the reported quantitative results, and these tasks are not applicable to this work. We have reviewed all AI-assisted work.

ETHICS STATEMENT

The proposed RI-LRP uses open-source LLMs publicly available on HuggingFace (Llama-2, Llama-3, and DeiT), and all datasets are public natural language and vision benchmarks and are publicly available (IMDB, ARC-Easy, Wikipedia, MATH-500, CodeJudge-Eval, SQuAD v2, and ImageNet-1K), which did not raise any ethical concerns.

REPRODUCIBILITY STATEMENT

The open-source code of the proposed method is accessible from the anonymous repository https://anonymous.4open.science/r/RI-LRP-6AF0.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197. Association for Computational Linguistics, July 2020. doi: 10.18653/v1/2020.acl-main.385.

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 *Proceedings of the 41st International Conference on Machine Learning*, pp. 135–168, 2024.

Ojobo Agbo Eje, Sayed Mehedi Azim, and Iman Dehzangi. Explainable ai applications in healthcare: A systematic review. *Algorithms*, 19(6):488, 2026.

Syed Ahmed, Bharathi Vokkaliga Ganesh, Jagadish Babu P, Karthick Selvaraj, Praneeth Talluri, Sanket Hingne, Anubhav Kumar, Anushka Yadav, Pratham Kumar Verma, Kiranmayee Janardhan, et al. Vista: Visualization of token attribution via efficient analysis. *Karthick and Talluri, Praneeth and Hingne, Sanket and Kumar, Anubhav and Yadav, Anushka and Verma, Pratham Kumar and Janardhan, Kiranmayee and AN, Mandanna, VISTA: Visualization of Token Attribution via Efficient Analysis (January 02, 2026)*, 2026.

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*, pp. 435–451. PMLR, 2022.

Fabrizio Angiulli, Fabio Fassetti, and Simona Nisticò. Improving local interpretable classifier explanations exploiting self-generated semantic features. *Neural Computing and Applications*, 37 (19):13617–13641, 2025.

Ido Andrew Atad, Itamar Zimerman, Shahar Katz, and Lior Wolf. Tensorlens: End-to-end transformer analysis via high-order attention tensors. *arXiv preprint arXiv:2601.17958*, 2026.

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):1–46, 2015.

Yarden Bakish, Itamar Zimerman, Hila Chefer, and Lior Wolf. Revisiting lrp: Positional attribution as the missing ingredient for transformer explainability. *Advances in Neural Information Processing Systems*, 38:88703–88735, 2026.

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*, pp. 342–350. PMLR, 2017.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.

Anju Chhetri, Jari Korhonen, Prashnna Gyawali, and Binod Bhattarai. Nero: Explainable out-of-distribution detection with neuron-level relevance. *arXiv preprint arXiv:2506.15404*, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

Varun Sunil Guwal, Reeba Khan, Anupama Jawale, Taher Kezar Jhalodwala, Shaad Abid Qureshi, Aditya Ojha, Hardik Malhotra, and Hardi Thakkar. Building trust in transformers with a shap-based approach for explaining ai decisions. In *2025 12th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1–6. IEEE, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Jianpeng Hu, Yanzeng Li, Jialun Zhong, Lei Zou, and Wenfa Qi. Detecting hallucinations in retrieval-augmented generation via semantic-level internal reasoning graph. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 27826–27841, 2026.

Qi Huang, Furong Ye, Ananta Shahane, Thomas Bäck, and Niki van Stein. From heuristic selection to automated algorithm design: Llms benefit from strong priors. *arXiv preprint arXiv:2603.02792*, 2026.

Lasse Marten Jantsch, Dong-Jae Koh, Seonghyeon Lee, and Young-Kyoon Suh. Dual path attribution: Efficient attribution for swiglu-transformers through layer-wise target propagation. *arXiv preprint arXiv:2603.19742*, 2026.

Jihang Li, Jiacheng Qiu, Yin-Ping Zhao, and Zeyi Wen. Tilia: Enhancing lime with decision tree surrogates. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1582–1592, 2025a.

Xinpeng Li and Kai Ming Ting. Distribution-based feature attribution for explaining the predictions of any classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23221–23229, 2026.

Yanshan Li, Huajie Liang, and Lirong Zheng. Wb-lrp: Layer-wise relevance propagation with weight-dependent baseline. *Pattern Recognition*, 158:110956, 2025b.

Hugo Touvron Llama. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2407.21783*, 2023.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Melkamu Abay Mersha and Jugal Kalita. Explainable ai: Context-aware layer-wise integrated gradients for explaining transformer models. *Neurocomputing*, pp. 133050, 2026a.

Melkamu Abay Mersha and Jugal Kalita. Explainable ai: Context-aware layer-wise integrated gradients for explaining transformer models. *Neurocomputing*, pp. 133050, 2026b.

Shweta Mishra. Tokenmizer: Graph-structured session memory for long-horizon llm context management. *arXiv preprint arXiv:2606.06337*, 2026.

Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *arXiv preprint arXiv:1512.02479*, 2015.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recogn.*, 65(C):211–222, May 2017. ISSN 0031-3203. doi: 10.1016/j.patcog.2016.11.008.

Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2501–2508, 2020.

Sisi Peng, Wenlin Zhang, Shunhang Li, and Dan Qu. Qr3ag: Dynamic retrieval-augmented generation via query-response relevance threshold judgement. *Expert Systems with Applications*, pp. 130087, 2025.

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

Muhammad Rashid, Elvio G Amparore, Enrico Ferrari, and Damiano Verda. Shapbpt: Image feature attributions using data-aware binary partition trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25099–25107, 2026a.

Muhammad Rashid, Elvio G Amparore, Enrico Ferrari, and Damiano Verda. Shapbpt: Image feature attributions using data-aware binary partition trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25099–25107, 2026b.

Farnoush Rezaei Jafari, Grégoire Montavon, Klaus-Robert Müller, and Oliver Eberle. Mambalrp: Explaining selective state space sequence models. *Advances in neural information processing systems*, 37:118540–118570, 2024.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Wojciech Samek, Leila Arras, Ahmed Osman, Grégoire Montavon, and Klaus-Robert Müller. Explaining the decisions of convolutional and recurrent neural networks. *Mathematical Aspects of Deep Learning*, 229, 2022.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Eight Suzuki, Hideitsu Hino, and Noboru Murata. Lig: Layer-wise integrated gradients for within-layer flow analysis in transformers. *arXiv preprint arXiv:2606.21564*, 2026.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Thanh Luong Tuan and Abhijit Sanyal. Ontology-constrained neural reasoning in enterprise agentic systems: A neurosymbolic architecture for domain-grounded ai agents. *arXiv preprint arXiv:2604.00555*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xinwei Wu, Heng Liu, Xiaohu Zhao, Yuqi Ren, Linlong Xu, Longyue Wang, Deyi Xiong, Weihua Luo, and Kaifu Zhang. Finding the translation switch: Discovering and exploiting the task-initiation features in llms. *arXiv preprint arXiv:2601.11019*, 2026.

Weiqiu You, Siqi Zeng, Yao-Hung Hubert Tsai, Makoto Yamada, and Han Zhao. When lrp diverges from leave-one-out in transformers. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 176–188, 2025.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Yuwei Zhao, Ziyang Luo, Yuchen Tian, Hongzhan Lin, Weixiang Yan, Annan Li, and Jing Ma. Codejudge-eval: Can large language models be good judges in code understanding? In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 73–95, 2025.

A MULTI-CLASS PROJECTION HEAD

The projection-head argument in the main text is stated for the binary case, where the softmax reduces to a function of the logit difference and a two-class common perturbation is exactly function-preserving. We now give the general C-class version. The conclusion is the same; the cleanest construction applies the perturbation to *all* classes, so that the softmax is invariant rather than merely renormalized.

Setup. Let $\mathbf{h}$ be the final hidden representation and let

$$z_c = \mathbf{h}^\top \mathbf{W}_c, \qquad c = 1, \dots, C,$$

with $\mathbf{W}_c$ the output weight of class c. Under the linear-propagation approximation of Achtibat et al. (2024), the unary relevance of the target class y is

$$\mathbf{R}_y(\mathbf{x}) \propto \mathbf{h} \odot \mathbf{W}_y,$$

which depends only on $\mathbf{W}_y$ and is blind to every competing direction $\mathbf{W}_k$, $k \neq y$.

All-class common perturbation. Add a common vector $\mathbf{u}$ to every class weight, $\mathbf{W}_c \mapsto \mathbf{W}_c + \mathbf{u}$ for all c. Then every logit shifts by the same scalar,

$$z'_c = \mathbf{h}^\top(\mathbf{W}_c + \mathbf{u}) = z_c + \mathbf{h}^\top \mathbf{u},$$

and the softmax is exactly invariant,

$$p'_c = \frac{e^{z_c + \mathbf{h}^\top \mathbf{u}}}{\sum_{c'} e^{z_{c'} + \mathbf{h}^\top \mathbf{u}}} = \frac{e^{z_c}}{\sum_{c'} e^{z_{c'}}} = p_c.$$

The same holds for every pairwise margin and for the ranking of all logits,

$$z'_y - z'_k = z_y - z_k \quad \text{for all } y, k,$$

so the model function, its predictive distribution, and its decision boundary are all exactly preserved.

Effect on unary attribution. The unary LRP attribution of the target logit, however, is not invariant:

$$\mathbf{R}_{y'}(\mathbf{x}) \sim \mathbf{h} \odot (\mathbf{W}_y + \mathbf{u}) = \mathbf{R}_y(\mathbf{x}) + \mathbf{h} \odot \mathbf{u}.$$

Standard LRP therefore assigns different relevance under two parameterizations that implement the same function, incorporating common-mode evidence that is irrelevant to the classification decision. This is the projection-head analogue of the function-preserving reparameterization control in Appendix B.

Remark on per-class perturbations. Modifying only the target and one competitor, $\mathbf{W}_y \mapsto \mathbf{W}_y + \mathbf{u}$ and $\mathbf{W}_k \mapsto \mathbf{W}_k + \mathbf{u}$, leaves the pairwise margin $z_y - z_k$ unchanged but does alter the softmax denominator, and hence the probabilities, whenever $C > 2$. The all-class construction above avoids this issue and gives an exactly function-preserving perturbation for any C. We therefore use it as the multi-class statement of the deficiency.

Contrastive seed. Under the same all-class perturbation, the contrastive target is exactly invariant:

$$\tilde{z}^{(k)} = z_{c_{(1)}} - \frac{1}{k-1}\sum_{i=2}^{k} z_{c_{(i)}} \;\longmapsto\; \tilde{z}^{(k)} + \Big(1 - (k-1)\cdot\frac{1}{k-1}\Big)\mathbf{h}^\top\mathbf{u} = \tilde{z}^{(k)}.$$

The common-mode component of the all-class perturbation therefore cancels exactly in the contrastive seed, whereas the unary seed shifts by $\mathbf{h}^\top\mathbf{u}$. This shows that the contrastive target removes the leading common-mode sensitivity to function-preserving reparameterizations; it does not remove sensitivity arising from the nonlinear ϵ-rule in deeper layers, which we quantify separately in Appendix B.

B FUNCTION-PRESERVING REPARAMETERIZATION CONTROL

This control isolates a structural property of the unary seed. We construct a reparameterization of a ReLU network that leaves every logit exactly unchanged for every input, yet alters the input relevance produced by the unary seed under a fixed ϵ-rule. Because a faithful attribution should depend only on the function implemented and not on the parameterization chosen to implement it, this control shows that the unary seed only partially satisfies implementation invariance. We report the same quantity for the contrastive seed to quantify how much of this sensitivity it removes.

Network. We use a two-layer ReLU network with

$$\mathbf{h} = \mathrm{ReLU}(W_1\mathbf{x}), \qquad \mathbf{z} = W_2\mathbf{h},$$

where $\mathbf{x} = (1, 2)^\top$, $W_1 = I_2 \in \mathbb{R}^{2\times 2}$, and

$$W_2 = \begin{pmatrix} 2 & 1 \\ 0.5 & 1.5 \end{pmatrix}.$$

Both layers have the same width, so relevance can be enumerated by hand.

Reparameterization. For any $\lambda > 0$, define

$$(W_1^{(\lambda)},\, W_2^{(\lambda)}) = (\lambda W_1,\; \lambda^{-1}W_2).$$

Since ReLU is positively homogeneous, $\mathrm{ReLU}(\lambda W_1\mathbf{x}) = \lambda\,\mathrm{ReLU}(W_1\mathbf{x}) = \lambda\mathbf{h}$, hence

$$\mathbf{z}^{(\lambda)} = W_2^{(\lambda)}\,\mathrm{ReLU}(W_1^{(\lambda)}\mathbf{x}) = \lambda^{-1}W_2 \cdot \lambda\mathbf{h} = W_2\mathbf{h} = \mathbf{z}.$$

Every logit is therefore identical for every input, and every training gradient with respect to the function is identical as well. The two parameterizations implement the same function f. We set $\lambda = 10$ for the reported numbers.

Propagation. Relevance is seeded at the output as $R_c = w_c z_c$, where $\mathbf{w}$ is the output seed defined below, and is propagated to the input with the standard positive-denominator ϵ-rule:

$$R_i = \sum_j \frac{a_i w_{ij}}{\sum_{i'} a_{i'} w_{i'j} + \epsilon}\, R_j, \qquad \epsilon = 0.1.$$

The stabilizer ϵ is essential here: it is what breaks scale equivariance, so the rule is *not* invariant under $(\lambda W_1, \lambda^{-1}W_2)$. With $\epsilon = 0$ the rule would be scale-equivariant and the control would be vacuous; we state this explicitly to preempt the objection.

Table 3: Function-preserving reparameterization control. Input relevance (R_1, R_2) under the original and rescaled parameterizations, for both seeds. All logits are identical in the two columns, so any change in relevance is attributable to the parameterization alone.

Seed	(W_1, W_2)	$(10W_1,\ W_2/10)$	ℓ_1 change
Unary	$(1.774,\ 1.858)$	$(1.932,\ 1.942)$	0.241
Contrastive $k = 2$	$(1.553,\ 0.469)$	$(1.691,\ 0.490)$	0.159

Seeds. We compare two output seeds on the same network:

- **Unary seed:** $\mathbf{w} = (1,\ 0)$, i.e. propagate z_1 alone.
- **Contrastive seed ($k = 2$):** $\mathbf{w} = (1,\ -\frac{1}{2})$, i.e. propagate $z_1 - \frac{1}{2}z_2$, matching the $k = 2$ RI-LRP target.

Results. Table 3 reports the input relevance (R_1, R_2) under both parameterizations. The unary seed's relevance changes by 0.241 in ℓ_1 norm, and the contrastive seed's by 0.159, even though the underlying function—and hence every logit and every function-level gradient—is unchanged. This confirms that the unary seed's attribution depends on the parameterization rather than solely on the function, and quantifies the extent to which the contrastive seed reduces, but does not remove, this dependence.

Generality. The effect is not specific to the chosen constants. For any seed whose output weights are not scale-balanced across the two layers, the ϵ-rule yields relevance that varies with λ; we choose $\epsilon = 0.1$ to make this variation clearly measurable while keeping the rule numerically stable. The control is thus a representative instance of a general sensitivity rather than a hand-picked coincidence. The two-layer, width-two network is the smallest configuration that exhibits both the common-mode shift and its attenuation by the contrastive seed; larger networks introduce additional nonlinear interactions that obscure the effect being demonstrated.

Extension to deep Transformer architectures. The mechanism above is not an artifact of the two-layer toy network. The same argument applies verbatim to any depth: for a positive-homogeneous activation ϕ (ReLU, SiLU above threshold, or the gated projection in a Transformer MLP), an entire block can be rescaled as $(W_1, W_2) \mapsto (\lambda W_1, \lambda^{-1} W_2)$ without changing the block's input–output map, and this rescaling can be propagated through the remaining layers so that the full Transformer implements the same function on every input. The LRP ϵ-rule, however, is applied at each of the L linear layers in the deep stack, and each application carries its own stabilizer ϵ. Since the rule is not scale-equivariant once $\epsilon > 0$, the perturbation introduced at every layer accumulates rather than cancels, so the resulting input relevance varies with λ even though all logits, attention distributions, and function-level gradients are exactly preserved. In this sense the toy network isolates the smallest instance of a behavior that is present in full Transformers as well; the contrastive seed attenuates the common-mode component of this variation at every layer, but does not eliminate it. This is consistent with the observation of You et al. (2025) that bilinear and stabilized LRP rules in attention layers are sensitive to implementation choices that leave the function unchanged.

C Algorithm Details

This section specifies the concrete LRP rules used by the abstract operators $\rho(\cdot)$ and $\mathrm{PosLRP}(\cdot)$ in Algorithm 1, following AttnLRP (Achtibat et al., 2024) and PE-LRP (Bakish et al., 2026), and then describes the algorithm in detail.

C.1 Elementary LRP rules

Let $R_j^{(l+1)}$ denote the relevance of neuron j in layer $l+1$, a_i the activation of neuron i in layer l, and w_{ij} the weight connecting i to j. Every rule redistributes $R_j^{(l+1)}$ to the lower layer so that $\sum_i R_{i\leftarrow j}^{(l)} = R_j^{(l+1)}$ up to the stabilizer term.

ϵ-rule (linear layers). For a dense linear layer $z_j = \sum_i a_i w_{ij}$,

$$R_{i\leftarrow j} = \frac{a_i w_{ij}}{\sum_{i'} a_{i'} w_{i'j} + \epsilon} R_j, \qquad \epsilon = 10^{-6}. \tag{1}$$

The stabilizer ϵ prevents division by zero when the pre-activation is vanishingly small, at the cost of a small conservation error.

Identity rule (activation functions). For elementwise positive-homogeneous activations ϕ (SiLU, ReLU, Sigmoid, and the SiLU gate in the SwiGLU projection),

$$R_{i\leftarrow j} = R_j \quad \text{whenever } i = j, \qquad 0 \text{ otherwise.} \tag{2}$$

Relevance passes through the nonlinearity unchanged, since these activations preserve the sign and relative magnitude of their input.

Uniform rule (elementwise multiplication in the gated projection). For the elementwise product $z_j = a_i \cdot b_i$ between a linear projection and its SiLU gate, relevance is split evenly between the two factors,

$$R^{(a)}_{i\leftarrow j} = \tfrac{1}{2} R_j, \qquad R^{(b)}_{i\leftarrow j} = \tfrac{1}{2} R_j. \tag{3}$$

This avoids attributing the gate's nonlinearity entirely to one factor and is the rule used for the projection–SiLU multiplication.

Bilinear rule (attention $QK^\top$ and AV). For a bilinear layer $z_j = \sum_i a_i b_i$ with two activation factors, we apply the uniform rule followed by the ϵ-rule,

$$R_{i\leftarrow j} = \frac{a_i b_i}{\sum_{i'} a_{i'} b_{i'} + \epsilon} R_j, \qquad \epsilon = 10^{-6}. \tag{4}$$

This is the sequential combination used for the attention value matrix multiplication and, with the softmax detached, for the query–key product.

Deep Taylor decomposition (softmax). For the softmax over attention logits and over output logits, relevance is redistributed by a first-order Taylor expansion at a root point,

$$R_{i\leftarrow j} = \frac{a_i w_{ij}}{\sum_{i'} a_{i'} w_{i'j}} R_j, \tag{5}$$

where the root is chosen so that the softmax output is zero. This is the Deep Taylor decomposition of Montavon et al. (2017).

C.2 Positional relevance rules

Following PE-LRP (Bakish et al., 2026), we reformulate the input space as token–position pairs and apply the following rules.

Input-level position encoding. For the addition $z_i = P'_i + E_i + \epsilon$, relevance is propagated to the positional embedding by the ϵ-rule,

$$\mathcal{R}(P'_i) = P'_i \cdot \frac{\mathcal{R}(z_i)}{P'_i + E_i + \epsilon}. \tag{6}$$

RoPE at the attention level. For the rotation matrices $R_{i,k}$ applied to queries and keys, relevance is distributed evenly between the two,

$$\mathcal{R}(R_{i,k}) = \tfrac{1}{2}\mathcal{R}(\tilde{Q}_i) + \tfrac{1}{2}\mathcal{R}(\tilde{K}_i). \tag{7}$$

Multi-sink aggregation. Semantic and positional relevance are aggregated by summing only positive contributions,

$$\mathcal{R}_i = \sum_d E_i[d]^+ + \sum_k \sum_{d'} P_{i,k}[d']^+. \tag{8}$$

This positive-only aggregation is what leaves the final heatmap almost nonnegative.

Table 4: Component-to-rule mapping for RI-LRP. The right column gives the equation defining each rule in Appendix C.1 and Appendix C.2.

Component	Rule	Reference
Linear layers (Q, K, V, O, MLP)	ϵ-rule, $\epsilon = 10^{-6}$	Eq. equation 1
SiLU, ReLU, Sigmoid	Identity rule	Eq. equation 2
Projection–SiLU gate	Uniform rule	Eq. equation 3
Attention $QK^\top$, AV	Bilinear rule	Eq. equation 4
Softmax (attention, output)	Deep Taylor decomposition	Eq. equation 5
Input-level position encoding	ϵ-rule for addition	Eq. equation 6
RoPE (attention)	Even split Q/K	Eq. equation 7
Positional aggregation	Positive-only multi-sink	Eq. equation 8
Output seed	Top-k differential target	Eq. equation 9

C.3 Competitive output target

Given the logits $\mathbf{z} = f(\mathbf{x})$, let $(c_{(1)}, \ldots, c_{(k)}) = \mathrm{argsort}(\mathbf{z})[:k]$ be the top-k indices. The propagation target is

$$\tilde{z}^{(k)} = z_{c_{(1)}} - \frac{1}{k-1} \sum_{i=2}^{k} z_{c_{(i)}}. \tag{9}$$

The predicted logit receives weight $+1$, each of the $k-1$ competitors receives weight $-1/(k-1)$, and all other logits receive weight 0. The coefficients sum to zero, so a common offset a applied to every selected logit leaves the target unchanged: common-mode evidence is eliminated exactly at the output seed.

Because the LRP redistribution rules above are linear in the output seed for a fixed input, the raw signed propagation satisfies

$$\mathbf{R}^{(k)}_{\mathrm{raw}}(\mathbf{x}) = \Phi_{\mathrm{LRP}}(\mathbf{x}, z_{c^*}) - \frac{1}{k-1} \sum_{i=2}^{k} \Phi_{\mathrm{LRP}}(\mathbf{x}, z_{c_{(i)}}), \tag{10}$$

so the competitive target can be implemented in a single backward relevance-propagation pass. The top-k selection and weighting add work only at the output head, while the Transformer propagation cost is unchanged.

C.4 Component-to-rule mapping

Table 4 lists the concrete rule assigned to each Transformer component in our implementation, matching AttnLRP (Achtibat et al., 2024) and PE-LRP (Bakish et al., 2026).

C.5 Algorithm walkthrough

We now describe Algorithm 1 line by line.

Lines 1–3 (forward pass and target construction). The model is run forward to obtain the logit vector $\mathbf{z} = f(\mathbf{x})$. The top-$k$ indices are extracted by argsort, and the competitive target $\tilde{z}^{(k)}$ is formed by Eq. equation 9. For $k = 1$ this reduces to $z_{c_{(1)}}$, recovering standard single-logit LRP; for $k > 1$ the one-$(k-1)$th of the summed competitor logits is subtracted.

Lines 4–5 (initialization). The output relevance is set to the scalar target, $R_{\text{out}} = \tilde{z}^{(k)}$. The positional accumulator $\mathbf{R}_{\text{pos}}$ is initialized to zero; it will collect positional contributions at every layer.

Lines 6–12 (layer-wise propagation). The loop descends from the last layer $M-1$ to the first. For each neuron q in layer $l+1$, the pre-activation $z_{iq} = a_i w_{iq}$ is formed for all incoming neurons i, and the redistribution ratio $s_{iq} = \rho(z_{iq})/(\sum_{i'} \rho(z_{i'q}) + \varepsilon)$ is computed with the rule $\rho(\cdot)$ assigned to that component in Table 4. Relevance is then pushed down by $R_{i \leftarrow q} = s_{iq} R_q$ and summed over q to give R_i.

Algorithm 1 RI-LRP: Relative Influence Layer-wise Relevance Propagation

Input: Sample $\mathbf{x}$, model f, hyperparameter k, rule function $\rho(\cdot)$, stabilizer ε
Output: Attribution vector $\mathbf{R}(\mathbf{x})$

1: Forward pass: $\mathbf{z} = f(\mathbf{x})$
2: Get top-k indices: $(c_{(1)}, \ldots, c_{(k)}) = \text{argsort}(\mathbf{z})[:k]$
3: Target: $\tilde{z} = z_{c_{(1)}} - \frac{1}{k-1}\sum_{i=2}^{k} z_{c_{(i)}}$ ▷ Eq. equation 9
4: Initialize output relevance: $R_{\text{out}} = \tilde{z}$
5: Initialize positional accumulator: $\mathbf{R}_{\text{pos}} = \mathbf{0}$
6: **for** layer $l = M - 1$ **downto** 1 **do**
7: **for** each neuron q in layer $l + 1$ **do**
8: $z_{iq} = a_i \cdot w_{iq}$ for all i
9: $s_{iq} = \rho(z_{iq}) / (\sum_{i'} \rho(z_{i'q}) + \varepsilon)$ ▷ Table 4
10: $R_{i \leftarrow q} = s_{iq} \cdot R_q$
11: **end for**
12: $R_i = \sum_q R_{i \leftarrow q}$
13: $\mathbf{R}_{\text{pos}} \mathrel{+}= \sum_i \text{PosLRP}(P_{i,l}, R_i; \varepsilon)$ ▷ Eqs. equation 6, equation 7
14: **end for**
15: $\mathbf{R} = (R_1, \ldots, R_d) + \mathbf{R}_{\text{pos}}$ ▷ Eq. equation 8
16: **return** $\mathbf{R}$

Line 13 (positional accumulation). At each layer, the positional contribution of token i is computed by $\text{PosLRP}(P_{i,l}, R_i; \varepsilon)$ using Eqs. equation 6 and equation 7, and accumulated into $\mathbf{R}_{\text{pos}}$. This makes explicit that RI-LRP's new component is the competitive output target, while positional handling is held identical to the PE-LRP baseline.

Lines 14–15 (aggregation and return). The final heatmap combines the semantic input relevance with the accumulated positional relevance, $\mathbf{R} = (R_1, \ldots, R_d) + \mathbf{R}_{\text{pos}}$, and is returned. The positive-only aggregation of Eq. equation 8 is nonlinear, so the linear decomposition of the raw signed propagation does not carry over to the final heatmap.

D Experimental Setup

D.1 Datasets and Experimental Configuration

We evaluate our method across a diverse suite of benchmarks, encompassing natural language understanding, reasoning, code understanding, and computer vision. The following describes the distribution and key parameters for each dataset.

- **IMDB.** For binary sentiment classification, we utilize the IMDB dataset Maas et al. (2011), which consists of 50,000 movie reviews split evenly into 25,000 training and 25,000 test samples. The dataset is balanced, with an equal number of positive and negative reviews. We fine-tune three variants of the Llama model using cross-entropy loss on a 7:3 train/evaluation split. Training is performed with a learning rate of 5×10^{-4} and 5 warmup steps. Both models converge to 90% accuracy within approximately 10 epochs.
- **ARC-Easy.** The AI2 Reasoning Challenge (ARC) Clark et al. (2018) is a multiple-choice science QA benchmark designed for grades 3 to 9. It comprises a Challenge set of 1,172 questions and an Easy set of 2,376 questions, totaling 3,548 samples. We evaluate on a frozen Llama-3.1-8B-Instruct model, mapping answer letters (A–D) to labels via a fixed prompt. Only samples where the predicted token matches a valid option are retained.
- **Wikipedia.** We use a randomly selected subset of 2,500 texts from the Wikipedia 20231101.en dump (HuggingFace `wikimedia/wikipedia`), evaluated with a frozen Llama-3.1-8B-Instruct model under next-token prediction. All inputs are truncated to a fixed length of 512 tokens.
- **MATH-500.** This dataset is a curated subset of 500 problems from the MATH benchmark Hendrycks et al. (2021). It covers multiple mathematical subjects, including algebra, geometry, number theory, and calculus. The input is formed by concatenating the problem

and solution, followed by the phrase "The answer is:", and is evaluated using a frozen Llama-3.1-8B-Instruct model.

- **CodeJudge-Eval.** For code understanding evaluation, we sample 1,000 instances from CodeJudge-Eval Zhao et al. (2025), a multiple-choice benchmark built from 5,000 APPS problems across three difficulty levels, with candidate solutions generated by 16 LLMs and verified via test-case execution to obtain fine-grained verdicts (e.g., Accepted, Wrong Answer, Runtime Error). Each sample is tokenized and fed to a frozen Llama-3.1-8B-Instruct model.
- **SQuAD v2.** We use 1,000 answerable questions from the SQuAD v2.0 development set Rajpurkar et al. (2018), a reading-comprehension benchmark of questions over Wikipedia paragraphs with annotated answer spans. Following the protocol in Appendix F, questions are drawn with seed 42 and filtered to those whose first gold answer aligns to context tokens and whose prompt fits within 512 input tokens. The attribution target is the first token of the first gold answer, evaluated with a frozen Llama-3.1-8B-Instruct model.
- **ImageNet.** We employ the ImageNet-1K dataset Russakovsky et al. (2015), a standard benchmark for large-scale image classification containing 1.28 million training images and 50,000 validation images across 1,000 classes. For our experiments, we randomly select 50 classes, each with 100 samples. We utilize DeiT models of varying sizes (tiny, small, base). Images are resized to 224×224 and normalized with ImageNet statistics. Saliency maps are computed at a patch resolution of 14×14 and upsampled via bilinear interpolation.

Common Hyperparameters. The random seed for sampling and algorithm is set to 42. Relevance tensors are normalized to $[0, 1]$ using min-max scaling: absolute-value min-max for text modalities and pure min-max for vision. Perturbation analysis covers both generation (most relevant first) and pruning (least relevant first), measured across 11 fractions from 0.0 to 1.0.

D.2 Implementation Details of Baselines

All experiments are conducted on servers equipped with Intel Xeon Gold 5218 CPUs and four NVIDIA A100 GPUs, running Ubuntu 20.04.6 with Linux kernel 5.4.0. The software environment includes CUDA 12.9 and Python 3.11. Key hyperparameters for each method are summarized below.

- **Input×Grad** Simonyan et al. (2013). Relevance is computed as $\sum_d \frac{\partial f}{\partial x_{t,d}} \cdot x_{t,d}$ throughout the network. This method serves as a simple yet effective baseline that captures local linear approximations of the model's decision boundary.
- **IG** Sundararajan et al. (2017). Uses $m = 20$ interpolation steps along a straight-line path from a zero baseline $x^0 = \mathbf{0}$ to the input x, with a Riemann left-sum approximation: $\mathrm{IG}(x) = (x - x^0) \odot \frac{1}{m} \sum_{k=0}^{m-1} \nabla F(x^0 + \frac{k}{m}(x - x^0))$. Final relevance is obtained by taking the absolute value of IG and summing over the embedding dimension. This method satisfies completeness and sensitivity axioms, providing theoretically grounded attributions that guarantee conservation of the model's output.
- **SHAP** Lundberg & Lee (2017). Tokens are partitioned into groups of size $g = 10$ according to attention mask order. The number of coalition samples is set to 200. Coalition weights follow the Shapley kernel: $\pi(z') = \frac{M-1}{\binom{M}{|z'|} \cdot |z'| \cdot (M - |z'|)}$, computed in log-space for numerical stability. Absent token groups are masked with zero embeddings. Shapley values are estimated via weighted least squares with Tikhonov regularization ($\lambda = 10^{-6}$), then uniformly distributed within each group. All forward passes are batched with size 16. No gradient computation is required, making this method applicable to black-box models where internal activations are inaccessible.
- **AttnRoll** Abnar & Zuidema (2020). The rollout matrix is iteratively computed with residual connections: $R \leftarrow 0.5 \cdot R \cdot \bar{A}_l + 0.5 \cdot I$, where $\bar{A}_l$ is the attention matrix averaged across all heads at layer l. Relevance is defined as the last row of the final rollout matrix. The rollout is performed over all layers of the model, with each layer's attention weights first normalized along the column dimension to ensure row-stochasticity.
- **CP-LRP** Ali et al. (2022). Applies stop relevance rule to all nonlinear activations (SiLU, Softmax), restricting relevance propagation exclusively to linear layers. This conservative

strategy ensures that relevance flows only along strictly linear paths, effectively isolating the contribution of attention mechanisms while suppressing non-linear interactions.

- **AttnLRP** Achtibat et al. (2024). Activations (SiLU, ReLU, Sigmoid) utilize the identity rule (relevance passes through unchanged); Softmax employs Deep Taylor decomposition; the element-wise multiplication in the projection-SiLU gate adopts the uniform rule; the bilinear attention value matrix multiplication applies a sequential combination of the uniform rule and the ϵ-rule with $\epsilon = 10^{-6}$; linear layers adopt the standard ϵ-rule with $\epsilon = 10^{-6}$. This configuration allows relevance to propagate through attention mechanisms in a principled manner while preserving the model's computational graph.
- **PE-LRP** Bakish et al. (2026). Built upon AttnLRP with position-encoding (PE) relevance propagation enabled. PE-LRP reformulates the input space as token-position pairs and introduces specialized LRP rules for PE layers. For input-level PE, relevance is propagated via the LRP-ϵ rule for addition: $\mathcal{R}(P'_i) = P'_i \cdot \frac{\mathcal{R}(z_i)}{P'_i + E_i + \epsilon}$. For attention-level PE (RoPE), relevance from queries and keys is distributed evenly to rotation matrices: $\mathcal{R}(R_{i,k}) = \frac{1}{2}\mathcal{R}(\tilde{Q}_i) + \frac{1}{2}\mathcal{R}(\tilde{K}_i)$. Semantic and positional relevance are aggregated via a multi-sink approach summing only positive contributions: $\mathcal{R}_i = \sum_d E_i[d]^+ + \sum_k \sum_{d'} P_{i,k}[d']^+$.
- **DFAX** Li & Ting (2026). Defaults to a kernel-density-based mode, where each token's L2 norm $\|e_t\|_2$ is scored as $s_t = \text{KDE}_{\text{target}}(\|e_t\|_2) - \text{KDE}_{\text{other}}(\|e_t\|_2)$, reflecting how much more typical the token's embedding magnitude is for the target class than for others. Class-wise KDE bandwidth is selected via Silverman's rule: $h = 0.9 \cdot \min(\hat{\sigma}, \text{IQR}/1.34) \cdot n^{-1/5}$. A two-pass architecture is employed: the first pass collects reference distributions from all samples; the second pass computes per-sample relevance via leave-one-out scoring.
- **RI-LRP** (ours). Our method inherits the LRP propagation rules for semantic embedding relevance: activations (SiLU, ReLU, Sigmoid) use the identity rule; Softmax employs Deep Taylor decomposition; the projection-SiLU gate element-wise multiplication adopts the uniform rule; the bilinear attention value matrix multiplication applies a sequential combination of the uniform rule and the ϵ-rule ($\epsilon = 10^{-6}$); linear layers follow the standard ϵ-rule ($\epsilon = 10^{-6}$). For positional relevance, RI-LRP propagates relevance through position encoding layers via the LRP-ϵ rule for input-level PE addition and evenly distributing relevance between queries and keys for RoPE at the attention level, with semantic and positional relevance aggregated via a multi-sink approach summing only positive contributions. The competitive window size k is determined via grid search on a log scale over 80 sampled points for each dataset. This search yields intuitive and interpretable choices: $k = 2$ on IMDB, $k = 5$ on ARC-Easy, $k = 1000$ on Wikipedia and MATH-500, and $k = 10$ on CodeJudge-Eval. The selected values differ across tasks and perturbation protocols, reflecting the different competitive structure of each task. A separate SQuAD gold-target control at $k = 1000$ further improves fixed-size answer-span localization under its stated prompt and target selection (Section F).

E Results on an ImageNet-1K Subset

To evaluate the faithfulness of attribution methods in the vision domain, we adopt two complementary perturbation protocols: positive and negative perturbation. In positive perturbation, pixels are progressively masked in descending order of their relevance scores. An effective explainer should cause a sharp decline in model accuracy as the most critical regions are removed. Conversely, negative perturbation masks pixels in ascending order of relevance, where a reasonable method should preserve prediction stability when uninformative regions are suppressed. Both protocols are quantified by the area under the activation curve.

Table 5 reports perturbation results on the 50-class ImageNet-1K subset across DeiT-Base, -Small, and -Tiny. RI-LRP obtains the best reported value in each configuration. On DeiT-Tiny, for example, it increases negative prediction accuracy from 70.469 (PE-LRP) to 71.337 and reduces positive prediction accuracy from 5.257 to 4.950. These results show that the contrastive target transfers to the vision setting.

Model	Method	Negative↑		Positive ↓	
		Pred	Tag	Pred	Tag
Base	Random	51.063	50.164	51.041	50.365
	CP-LRP	68.958	67.631	28.195	26.178
	AttnLRP	76.178	74.626	18.574	18.414
	PE-LRP	76.494	75.049	18.796	18.587
	RI-LRP(ours)	**76.923**	**75.821**	**18.302**	**18.219**
Small	Random	44.340	43.192	44.700	43.412
	CP-LRP	70.450	67.760	16.890	16.483
	AttnLRP	73.675	70.706	13.925	13.546
	PE-LRP	74.521	71.365	14.153	13.777
	RI-LRP(ours)	**75.148**	**72.722**	**13.780**	**13.535**
Tiny	Random	28.494	27.706	28.766	28.024
	CP-LRP	32.461	31.362	14.934	19.870
	AttnLRP	52.185	47.516	10.784	8.032
	PE-LRP	70.469	65.796	5.257	5.024
	RI-LRP(ours)	**71.337**	**66.796**	**4.950**	**4.769**

Table 5: Comparison across DeiT model sizes on the 50-class ImageNet-1K subset under positive and negative perturbation. Negative metrics are accuracies after masking least-relevant pixels (higher is better), whereas positive metrics are accuracies after masking most-relevant pixels (lower is better). Pred and Tag denote predicted- and ground-truth-label targets. Bold indicates the best result in each group.

Table 6: Gold-target SQuAD v2 answer localization on the same 1000 questions. Top-1 is answer-span hit rate, not question-answering accuracy. $\mathrm{IoU}_{|G|}$ uses the top $|G|$ context tokens. Higher is better. Bold indicates the best result in each column.

Method	Top-1	$\mathrm{IoU}_{\lvert G\rvert}$
AttnLRP-style (unary)	0.405	0.262
PE-LRP (unary)	0.475	0.285
RI-LRP ($k = 1000$)	**0.677**	**0.349**

F SQuAD v2 Answer-Span Localization

Following the SQuAD plausibility motivation of Achtibat et al. (2024), we reuse the 1000 answerable development questions drawn with seed 42. We permuted the development questions, retained examples whose first annotated answer aligned to context tokens and whose prompt fit within 512 input tokens, and took the first 1000 eligible examples. The prompt places the question and context before an answer cue; the attribution target is the *first token of the first gold answer*. All three methods use the same model and question IDs. AttnLRP-style attribution propagates a unary semantic seed, PE-LRP adds positional relevance to that unary seed, and RI-LRP adds the same positional relevance to the target contrasted with the 999 highest-logit alternatives at $k = 1000$. The unary rows use the matching detached-logit seed, so that the competitive window is the principal difference between PE-LRP and RI-LRP.

We rank context tokens by signed final relevance in descending order, without taking absolute values. Top-1 is the fraction of examples for which the highest-ranked token overlaps the annotated answer. For each answer-token set G, $\mathrm{IoU}_{|G|}$ compares G with the top $|G|$ ranked context tokens. This test uses the annotated answer to choose the output token and measures answer-span localization, complementing the positive-mask IoU on correctly answered questions in Achtibat et al. (2024). The AttnLRP-style row uses the semantic path of our shared custom model.

Relative to unary PE-LRP, RI-LRP improves Top-1 by 0.202 (paired 95% bootstrap interval $[0.178, 0.228]$) and $\mathrm{IoU}_{|G|}$ by 0.0649 ($[0.0530, 0.0770]$). These gains describe the specified gold-target prompt and ranking rule, and together with the signed predicted-token margin control give a consistent picture of the advantage on answer-span localization.

Table 7: Wikipedia next-token rank diagnostics ($n = 1{,}000$). Gaps are in logit units. Tail mass is measured against the full vocabulary; "tail > winner" means $T_i(k) > p_{i,(1)}$. The $k = 64$ row is a descriptive probability-curve elbow under a search budget of 1,000 matching the prior experiment; $k = 1000$ is the value selected by the paper's validation AUAC.

k	Median $g_i(k)$ [IQR]	Mean $T_i(k)$	Median $T_i(k)$	$T_i(k) > 0.05$	Tail > winner
64	8.34 [6.62, 9.94]	0.0614	0.0265	32.8%	9.0%
1000	12.06 [10.63, 13.97]	0.0186	0.0084	8.7%	2.5%

G LOGIT-RANK AND TAIL DIAGNOSTICS FOR THE TOP-k SET

The top-k output seed defined in the main text selects competitors by logit rank, whereas the choice of k is made by validation AUAC. To examine what is excluded by this selection, we analyze an unlabeled diagnostic sample of 1,000 nonempty articles drawn without replacement from the 64,078-article local Wikipedia dataset (seed 42). We evaluate Llama-3.1-8B-Instruct at the next-token position after up to the first 512 tokens of each article, one article per forward pass, with temperature-one softmax over the full 128,256-token vocabulary.

Write $z_{i,(r)}$ and $p_{i,(r)}$ for the rank-r logit and full-vocabulary softmax probability on context i. The winner-to-rank gap and omitted probability mass are

$$g_i(r) = z_{i,(1)} - z_{i,(r)} = \log \frac{p_{i,(1)}}{p_{i,(r)}}, \qquad T_i(k) = 1 - \sum_{r=1}^{k} p_{i,(r)}. \tag{11}$$

Thus the gap is recoverable exactly from ranked softmax probabilities at temperature one; $T_i(k)$ includes *all* omitted vocabulary items, including ranks beyond the 4,096 probabilities saved per context. We also report $a_i(k) = \log[p_{i,(1)}/T_i(k)]$, the logit margin between the winner and the *aggregate* outside-set hypothesis $\log \sum_{r>k} \exp z_{i,(r)}$. A positive $a_i(k)$ means that the winner alone has more probability than the entire omitted tail.

At the validation-selected $k = 1000$, the median winner-to-cutoff gap is 12.06 logits, corresponding to a cutoff-to-winner probability ratio of approximately $e^{-12.06} = 5.8 \times 10^{-6}$. Since every excluded token has rank greater than 1000, its individual probability relative to the winner is no larger than that of the cutoff token. The mean omitted mass is 1.86% (article-bootstrap 95% interval $[1.66\%, 2.05\%]$), and the median is 0.84%. The aggregate outside-set mass nevertheless exceeds 5% in 8.7% of contexts and exceeds the winner's probability in 2.5%. These exceptions matter: a large per-token logit gap does not imply that the sum over a large omitted vocabulary is always negligible. If the softmax itself is renormalized only over the top 1000 tokens, the increase in the winner's log probability is exactly $-\log[1 - T_i(1000)]$; its median is 0.0084 nats and its 95th percentile is 0.0741 nats. Similarly, for the full-vocabulary log-softmax objective, the sum of absolute output-gradient coefficients on omitted logits is exactly $\sum_{r>k} |\partial \log p_{i,(1)}/\partial z_{i,(r)}| = T_i(k)$. These identities quantify probability-level truncation at the observed logits; they do not bound changes in input attribution, for which the logit Jacobians and LRP redistribution also matter.

A probability-based elbow provides an independent descriptive scale. Let $\bar{C}(k) = 1000^{-1} \sum_i [1 - T_i(k)]$ and maximize its normalized distance from the line joining $k = 1$ and $k = 1000$. This yields $k = 64$ (article-bootstrap 95% interval $[61, 67]$), at which the mean covered probability is 93.86%; the corresponding value at $k = 1000$ is 98.14%. The elbow depends on the rank budget: budgets of 256, 512, 2048, and 4096 give knees of 25, 40, 105, and 171, respectively. The $k = 64$ elbow is therefore a distributional diagnostic, not a replacement for validation-based selection or evidence that 64 maximizes attribution faithfulness.

Scope of the top-k approximation. The measurements support the *local ranking* premise that, at the unperturbed next-token position, the omitted logits are individually well below the winner for most contexts. They do not establish that their total influence on the paper's uniformly weighted RI-LRP seed is negligible. In fact, for the shift-invariant mean competitor gap $\bar{g}_i(k) = (k-1)^{-1} \sum_{r=2}^{k} g_i(r)$, the median increases from 6.84 at $k = 64$ to 10.77 at $k = 1000$; the median paired increase is 3.90 logits. The reported seed satisfies $\tilde{z}_i^{(k)} = z_{i,(1)}/k + (k-1)\bar{g}_i(k)/k$, so changing the set can

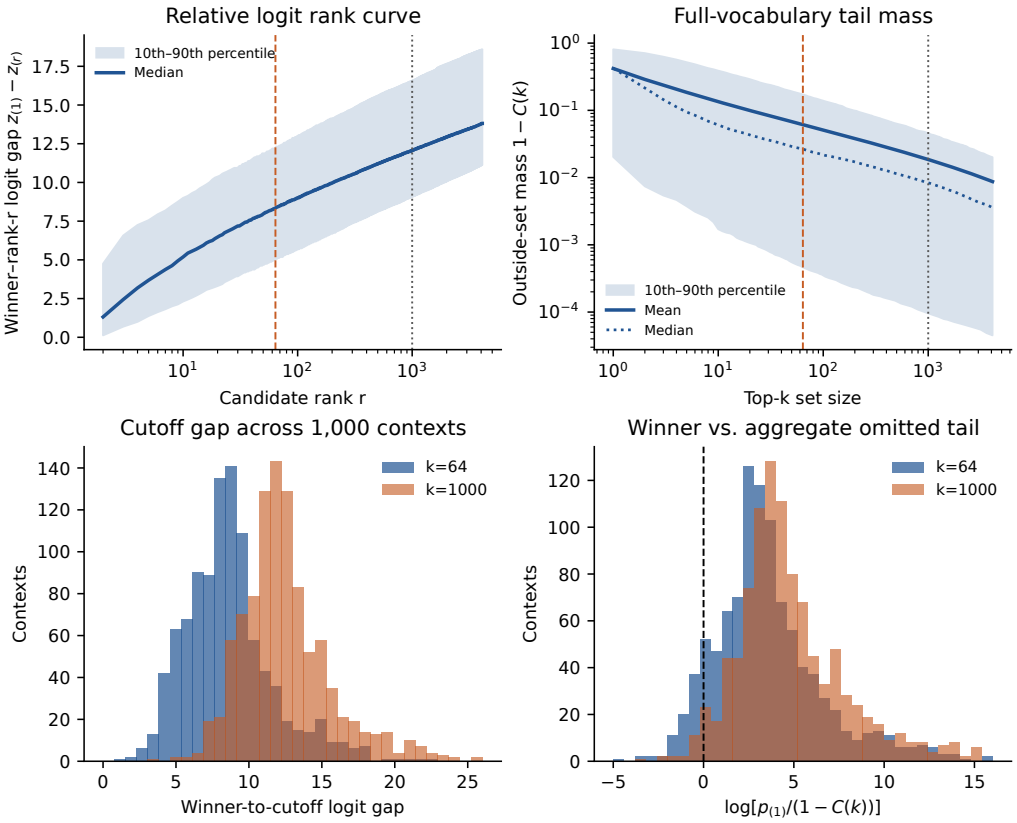

Figure 6: Wikipedia next-token output distributions on the same 1,000 random contexts. Top left: winner-to-rank logit gap, with median and 10th–90th percentile band. Top right: full-vocabulary probability outside the top-k set, with mean, median, and percentile band. Vertical lines mark $k = 64$ (dashed) and $k = 1000$ (dotted). Bottom: per-context distributions of the cutoff gap and winner-versus-aggregate-tail log odds; values below zero in the latter indicate an aggregate omitted tail with more mass than the winner.

materially change the scalar being propagated even when the additional tokens have low softmax probability. Nor do unperturbed ranks bound how far logits can move after an input perturbation. We therefore treat top-k as a validated, finite competitive window, rather than as a theorem that the excluded logit tail has zero discriminative or attribution effect. Figure 4 reports the task-level k sweep; Table 7 adds the distributional evidence and its limits.

H Calculation time on three datasets of different lengths

Fig. 7 reports the wall-clock runtime of all ten methods on 50 samples from each of three datasets with different average token counts: MATH-500 (average 274 tokens), Wikipedia (average 512 tokens), and CodeJudge-Eval (average 850 tokens). Each method is run ten times on the same samples, and the box plots show the resulting distribution; the vertical axis is on a log scale.

The runtime of all methods increases with input length, but the rate of increase differs substantially across method families. Input×Grad is the fastest method in every panel, staying below 12 seconds even on CodeJudge-Eval. AttnRoll and CP-LRP follow, with CP-LRP slightly faster than the full LRP variants. AttnLRP, PE-LRP, and RI-LRP form a tight cluster whose runtimes are close to each other in all three panels; RI-LRP remains within a narrow margin of AttnLRP and PE-LRP, confirming that the contrastive output seed adds no measurable propagation overhead beyond the top-k selection at the output head. The gap between this cluster and the faster baselines grows only mildly with input length.

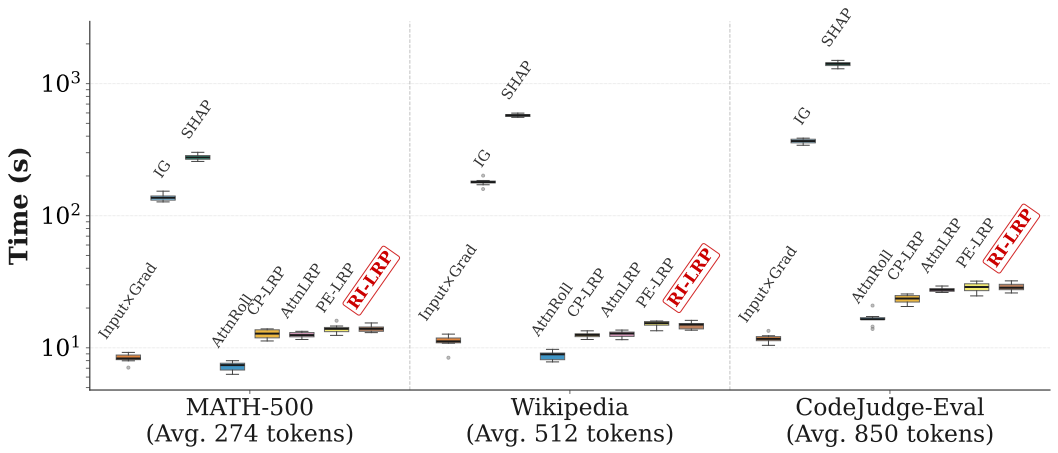

Figure 7: The runtime (log-scale) of the methods on 50 samples of three datasets with different number of tokens. Variance represents the deviation of running repeated 10 times.

IG and SHAP are one to two orders of magnitude slower than the LRP family. IG grows from roughly 1.4×10^2 seconds on MATH-500 to about 3.7×10^2 seconds on CodeJudge-Eval, while SHAP grows from about 2.8×10^2 to above 1.3×10^3 seconds. This scaling reflects their reliance on multiple forward or backward passes per sample, whose cost grows with the number of tokens.

Across all three input lengths, RI-LRP matches the runtime of the strongest LRP baselines while outperforming them in the perturbation and localization metrics reported in the main text. The efficiency gap relative to IG and SHAP widens as inputs grow longer, since those methods require many repeated passes per sample whereas RI-LRP requires a single relevance-propagation pass.