Improving LLM Attributions with Randomized Path-Integration

Oren Barkan[*1] **Yehonatan Elisha**[*2] **Yonatan Toib**[*1]
Jonathan Weill[2] **Noam Koenigstein**[2]

[1]The Open University, Israel
[2]Tel Aviv University, Israel

Abstract

We present Randomized Path-Integration (RPI) - a path-integration method for explaining language models via randomization of the integration path over the attention information in the model. RPI employs integration on internal attention scores and their gradients along a randomized path, which is dynamically established between a baseline representation and the attention scores of the model. The inherent randomness in the integration path originates from modeling the baseline representation as a randomly drawn tensor from a Gaussian diffusion process. As a consequence, RPI generates diverse baselines, yielding a set of candidate attribution maps. This set facilitates the selection of the most effective attribution map based on the specific metric at hand. We present an extensive evaluation, encompassing 11 explanation methods and 5 language models, including the Llama2 and Mistral models. Our results demonstrate that RPI outperforms latest state-of-the-art methods across 4 datasets and 5 evaluation metrics. Our code is available at: https://github.com/rpiconf/rpi

1 Introduction

Recent advancements in AI research have impacted numerous application domains, fueling innovation and progress in user modeling and personalization (Barkan and Koenigstein, 2016; He et al., 2017; Ben-Elazar et al., 2017; Wang et al., 2019; He et al., 2020; Barkan et al., 2019a, 2020a,b,d, 2021d, 2023f; Katz et al., 2022), natural language processing (NLP) (Mikolov et al., 2013; Vaswani et al., 2017; Devlin et al., 2018; Brown et al., 2020; Barkan, 2017; Barkan et al., 2020f,e, 2021b; Ginzburg et al., 2021; Malkiel et al., 2020, 2022b), computer vision (He et al., 2016; Dosovitskiy et al., 2020; Liu et al., 2022; Carion et al., 2020; Assran et al., 2023; Barkan et al., 2023e), and sound synthesis (Engel et al., 2020; Kong et al., 2020; Barkan and Tsiris, 2019; Barkan et al., 2019b, 2023g).

Within the field of NLP, the advent of Transformers (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2019) have ushered in a new era of complex language model (LM) architectures (Liu et al., 2019; Raffel et al., 2019; Brown et al., 2020; Touvron et al., 2023). These models, with their expanding size and complexity, have become integral components across diverse applications.

This surge in model capacity and complexity has intensified the need for a profound understanding of the decision-making processes intrinsic to LMs. However, the inherent complexity often shrouds the transparency of these models' predictions, prompting the development of explainability methods to unveil the contributing factors influencing their outputs. Consequently, a multitude of explanation methods has been devised (Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg and Lee, 2017; Abnar and Zuidema, 2020; Modarressi et al., 2023).

Simultaneously, with the evolution of explanation methods, there has been a proliferation of various explanation metrics designed for the quantitative assessment of explainability methods (Samek et al., 2017; DeYoung et al., 2020). However, achieving consensus within the literature on the ultimate explanation metric remains elusive, as each metric offers unique insights into different facets of explanation quality.

Acknowledging the diversity in explanation metrics, we introduce Randomized Path-Integration (RPI), a method designed to elucidate predictions made by LMs through the integration of internal attention scores and their gradients along a random path. The introduction of randomness into the integration path stems from the representation of the baseline as a random tensor drawn from a

[*] Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 9430–9446
November 12-16, 2024 ©2024 Association for Computational Linguistics

specified baseline distribution. As a result, RPI produces a variety of baselines from this distribution, forming a pool of candidate attribution maps. The versatility afforded by multiple baselines facilitates the selection of the most effective attribution map, contingent upon the evaluation metric under consideration.

Through extensive evaluation spanning a variety of explanations methods, LMs, and datasets, RPI exhibits superior performance compared to current state-of-the-art methods across multiple explanation metrics.

2 Related Work

Explainable AI includes a wide array of methods, all aimed at improving the understanding of decisions made by deep learning models across multiple application domains (Fong et al., 2019; Simonyan et al., 2013; Fong and Vedaldi, 2017; Selvaraju et al., 2017; Zhou et al., 2018; Barkan et al., 2020c, 2021c,a, 2023d,c,a,b; Gaiger et al., 2023; Barkan et al., 2024; Malkiel et al., 2022a; Chefer et al., 2021a,b; Sanyal et al., 2022).

Perturbation-based techniques such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) introduce perturbations to individual inputs or neurons and observe their consequential effects on subsequent neurons in the network.

Relevance-Decomposition methods look at the network's representation as vectors, each exerting distinct influences on the model's predictions. GlobEnc (Modarressi et al., 2022) and ALTI (Ferrando et al., 2022) incorporate local-level decomposition, aggregating resulting vector norms using rollout (Abnar and Zuidema, 2020) to construct global-level explanations. Recently, DecompX (Modarressi et al., 2023) was introduced, providing state-of-the-art results by constructing decomposed token representations and sequentially propagating them through the model without interlayer mixing.

Gradient-based methods rely on the gradients of the model's prediction score concerning the input tokens. Basic gradient methods, such as Vanilla Gradients (Simonyan et al., 2013) and GradientXInput (Shrikumar et al., 2016), operate on the principle of gradient computation. Integrated Gradients (IG) (Sundararajan et al., 2017) is a path-integration method, computing gradients on interpolated points along a straight line between the data and an uninformative baseline. Additional approaches such as DeepLift (Shrikumar et al., 2017) and GradientShap (Lundberg and Lee, 2017) can be viewed as approximations of IG. Other path-integration methods, such as Discretized Integrated Gradient (DIG) (Sanyal and Ren, 2021), treat the integration path differently. DIG replaces the continuous straight path with a discretized one, where interpolated points are words. Another innovative path-integration approach comes in the form of Sequential Integrated Gradients (SIG) (Enguehard, 2023). SIG addresses concerns about altered sentence meaning by computing the importance of each word while keeping other words fixed and creating interpolations between the baseline and the word of interest. Notably, SIG has demonstrated superior performance when compared to several existing methods, including DIG, IG, and GradientShap.

RPI complements this line of path-integration methods by modeling the baseline representation as a random tensor sampled from a Gaussian diffusion process (Sohl-Dickstein et al., 2015), and integrating the attention scores and their gradients along a randomized path between a randomly drawn baseline and the attention scores to form an attribution map. By baseline resampling from the baseline distribution, RPI establishes a pool of candidate attribution maps that facilitates the selection of the most effective attribution map based on the metric under consideration. Our evaluation indicates that this unique feature of RPI leads to state-of-the-art results and can be further incorporated to existing path-integration to enhance their performance.

3 Randomized Path-Integration

Let $\mathcal{V}$ denote the vocabulary, and let $u = (u_i)_{i=1}^k$ represent a sequence of k tokens constituting the input text, where each $u_i \in \mathcal{V}$. We further define $\mathbf{x}_u = [\mathbf{x}_{u_1}, ..., \mathbf{x}_{u_k}]$ as a 2D tensor in $\mathbb{R}^{d \times k}$ accommodating k embeddings, with each $\mathbf{x}_{u_i} \in \mathbb{R}^d$ representing the token u_i.

In the context of this work, our emphasis lies in the domain of classification tasks. Hence, we introduce a model denoted as F, designed to take an input $\mathbf{x}_u$ and yield an output $F(\mathbf{x}_u) \in [0, 1]^c$. Here, $F_i(\mathbf{x}_u)$ denotes the probability assigned to class i, and c signifies the total number of available classes. The model F can assume the form of a LM utilizing a classification head, producing probabilities for each class within the specific task. This process involves either finetuning the LM or utilizing it as a foundational component for the transfer learning of a specific task.

Recent LM architectures, often referred to as large language models (LLMs), are predominantly decoder-based, tasked with completing sequences of tokens in the output (Brown et al., 2020). Although these LLMs can be finetuned using classification heads, a more prevalent approach is classification via completion (Raffel et al., 2019). In this method, the LLM is prompted with the specific task, either in a few-shot or zero-shot scenario, and instructed to generate a token that signifies the correct class. For instance, in sentiment analysis, if the relevant classes are communicated to the LLM (via prompt) as 'positive' and 'negative', logit scores for the tokens 'positive' and 'negative' are computed, and softmax is applied to obtain the probabilities associated with each class. It is noteworthy that this approach can be implemented either before or after finetuning the LLM on the relevant dataset in a completion manner.

In this work, we conduct experiments in various settings, including finetuning with classification heads on top of the LLM as well as classification via completion (decoder-based models).

3.1 Integrated Gradients

IG (Sundararajan et al., 2017) enables the creation of an attribution map by defining a linear path between a baseline representation $\mathbf{b} \in \mathbb{R}^{d \times k}$ and $\mathbf{x}_u$ via the parameterization:

$$\mathbf{v}_u = \mathbf{b} + a(\mathbf{x}_u - \mathbf{b}) \text{ with } a \in [0, 1], \quad (1)$$

and accumulating the gradients along this path as follows:

$$\begin{aligned} \mathbf{m}_{IG} &= \int_0^1 \frac{\partial F_y}{\partial \mathbf{v}_u} \circ \frac{\partial \mathbf{v}_u}{\partial a} da \\ &\approx \frac{\mathbf{x}_u - \mathbf{b}}{n} \circ \sum_{j=1}^{n} \frac{\partial F_y}{\partial \mathbf{v}_u}, \end{aligned} \quad (2)$$

where $\circ$ denotes the Hadamard product, y denotes the class receiving the highest score in the prediction, and the approximation in the last transition is obtained by setting $a = \frac{j}{n}$ in Eq. 1. The resulting attribution map $\mathbf{m}_{IG}$ can be manipulated in various ways to form attribution scores for each individual element in $\mathbf{x}_u$.

3.2 The RPI method

RPI facilitates two distinctive features that set it apart from IG. Firstly, RPI integrates on the internal attention scores (rather than the token representation themselves), computed in the intermediate layers of F, and their gradients. This design allows for the aggregation of information from various attention heads and model layers w.r.t. each individual token in the input. Secondly, RPI models the baseline representation[1] $\mathbf{b}$ as a random tensor drawn from a distribution $\mathcal{B}$. This approach facilitates the sampling of multiple baselines, resulting in various integration paths, each leading to a distinct attribution map. Subsequently, one can select the attribution map associated with the integration path that yields the best results on the specific explanation metric under consideration. In what follows, we describe RPI in detail.

Let $\mathbf{a}_u^l \in R^{h \times k \times k}$ denote the attention scores tensor accommodating the h attention matrices produced by the l-th layer in the model F (when applied to a specific input $\mathbf{x}_u$). With a budget of trials R, the RPI process unfolds as follows: First, we sample R baselines $B^l = \{\mathbf{b}^{lr}\}_{r=1}^{R}$ from the baseline distribution $\mathcal{B}$, where $\mathbf{b}^{lr} \in R^{h \times k \times k}$. Accordingly, we redefine the interpolant for the attention tensor as:

$$\mathbf{v}_u^{lr} = \mathbf{b}^{lr} + a(\mathbf{a}_u^l - \mathbf{b}^{lr}) \text{ with } a \in [0, 1]. \quad (3)$$

Using B, we compute corresponding set of attribution maps $M^l = \{\mathbf{m}^{lr}\}_{r=1}^{R}$, where

$$\mathbf{m}^{lr} = \phi \left(\frac{\mathbf{a}_u^l - \mathbf{b}^{lr}}{n} \circ \sum_{j=1}^{n} \frac{\partial F_y}{\partial \mathbf{v}^{lr}} \circ \mathbf{v}^{lr} \right), \quad (4)$$

and ϕ is a function that takes the resulting tensor from the integration process, performs mean reduction on its first dimension (the attention heads dimension), and extracts a specific row from the resulting 2D matrix, producing the d-dimensional attribution map $\mathbf{m}^{lr}$. In encoder-based models, the

[1] Note that now the baseline representation $\mathbf{b}$ should match the dimensions of the attention tensor, which differ from dimensions of the input tensor dimensions.

extracted row is typically the one corresponding to the attention scores of the first token (e.g., in BERT, it is associated with the CLS). In decoder-based models, the extracted row is the last one corresponding to the last token used to generate the next token predicted by the model as part of the completion task[2].

We further note that the integrand in Eq.4 involves the Hadamard product of the interpolated attention tensor with its gradient. This is another difference from IG (Eq.2) that integrates the gradients only. We found that the multiplication by the interpolated attention tensor further improves the results, as it allows the amplification of elements in which both the gradient and the interpolated attention score are high. This design choice is further supported by the findings from (Selvaraju et al., 2017).

Finally, in the resulting attribution map $\mathbf{m}^{lr} \in \mathbb{R}^d$ (Eq. 4), the i-th entry $\mathbf{m}_i^{lr}$ represents the attribution score assigned to the token u_i (which is the i-th token in the input u).

Next, we describe the attribution map selection process. Let J be a set containing the indexes of layers in F participating the RPI process, and define the unified set of the resulting attribution maps by $M^J = \cup_{l \in J} M^l$. Consequently, the RPI attribution map is determined by:

$$\mathbf{m}_{\text{RPI}} = \underset{\mathbf{m} \in M^J}{\operatorname{argmax}}\, \psi(\mathbf{m}), \quad (5)$$

where ψ represents the metric under consideration[3], and $\psi(\mathbf{m})$ denotes the metric score on the attribution map $\mathbf{m}$. Ultimately, for simplicity, in this work, we opt for applying RPI with $J = \{L\}$, denotes the index of the last layer in F. As demonstrated in Sec. 4.2, this setup consistently produces state-of-the-art results. This setup further aligns with established explanation methodologies (Selvaraju et al., 2017; Caron et al., 2021) that predominantly utilize the final layer of the model for explanation generation. However, it is essential to note that RPI offers the flexibility to utilize all model layers and their combinations (Eq. 5). Therefore, in the Appendix (SectionA.1), we conduct an exhaustive ablation study, exploring the application of RPI across various layers and their aggregations.

3.2.1 The baseline distribution

In their study (Sturmfels et al., 2020), the authors investigated various baseline representations, without establishing a clear preference for any specific method. Consequently, a reasonable approach would be to define $\mathcal{B}$ as a mixture of distributions accommodating diverse baseline representations, each with its associated weight. In this work, we propose a more simple approach where the baseline is drawn from a Gaussian diffusion process (Sohl-Dickstein et al., 2015). Specifically, we define the baseline distribution at timestamp t as $\mathcal{B}_t = \mathcal{N}(\sqrt{\overline{\alpha}_t}\mathbf{a}_u^l, (1-\overline{\alpha}_t)\mathbf{I})$, where $\overline{\alpha}_t = \prod_{j=1}^t \alpha_j$, $\alpha_t = 1 - \beta_t$, and $\{\beta_t\}_{t=1}^T$ are parameters defining the variance schedule over T timesteps. Subsequently, each baseline of the R baselines in B^l is sampled by uniformly selecting timestamp t from $\{1, .., T\}$ and then drawing a baseline $\mathbf{b}^{lr}$ from $\mathcal{B}_t$.

The rationale for employing the Gaussian diffusion process to model the baseline as a random tensor is to establish a baseline distribution capable of accommodating diverse levels of noise, contingent upon the timestamp t. Specifically, this approach enables the introduction of small Gaussian noise for smaller values of t, gradually transitioning to higher levels of noise that ultimately converge to the standard normal noise as t increases.

Complexity The computational complexity of IG grows linearly with n (number of interpolations), while RPI introduces an additional factor, the number of sampled baselines R, as each baseline entails a different integration path. Fortunately, both IG and RPI are embarrassingly parallel. Therefore, given a GPU capable of accommodating a batch size of nR examples, IG and RPI are anticipated to exhibit comparable runtime performance in practical scenarios. Moreover, our empirical evaluation indicates that, even with a relatively small R, RPI consistently outperforms the latest state-of-the-art explanation methods across a variety of metrics, models, and datasets. Lastly, it is crucial to recognize that explanations serve primarily for debugging and auditing purposes, rather than real-time decision-making. In these contexts, prioritizing the generation of a higher-quality explanation over

[2] Recall that in this work, decoder-based models are customized for a classification task by expecting them to predict the correct class with the first predicted token in the completion.

[3] For a metric that favors lower scores, the operator in Eq. 5 should be changed to argmin.

speed is often considered advantageous.

4 Experimental Setup and Results

All experiments were conducted on a high-performance NVIDIA DGX server, equipped with 8 A100 GPUs, utilizing the PyTorch platform.

4.1 Experimental setup

Datasets and Models In our comprehensive evaluation, we assess various explanation methods across four distinct datasets, aiming to provide a thorough understanding of their efficacy in diverse scenarios characterized by different classification tasks and text lengths. **SST2** (Socher et al., 2013): Binary sentiment classification focusing on short texts. Rotten Tomatoes (**RTN**) (Pang and Lee, 2005): Binary sentiment classification in medium-sized texts. Emotion Recognition (**EMR**) (Saravia et al., 2018): Emotion classification with six classes (Sadness, Joy, Love, Anger, Fear, and Surprise) predominantly in short texts. AG News (**AGN**) (Zhang et al., 2015): Topic classification with four classes (World, Sports, Business, Sci/Tech) across texts of varying lengths.

Our evaluation involves five distinct model architectures: BERT-Base (Devlin et al., 2018), BERT-Base (Devlin et al., 2018), DistilBERT-Base (Sanh et al., 2019), Llama2 7B (Touvron et al., 2023) and Mistral 7B (Albert Q. Jiang, 2023). For the first three models, we utilize their finetuned versions tailored to each dataset. On the other hand, Llama2 and Mistral were evaluated in few-shot prompting mode without fine-tuning for the RTN and SST2 tasks, which aligns with their typical application in large language models (LLMs), as detailed in Sec. 3. For the AGN and EMR tasks, we finetuned the models using LoRA due to their limited performance in the few-shot prompting scenario. The prompts utilized in this research are presented in the Appendix. The processing of datasets, as well as the retrieval of both pretrained and finetuned versions of models for each dataset, was conducted using the HuggingFace library (Wolf et al., 2019). Our data processing approach closely follows the methodology outlined in (Enguehard, 2023). The complete code for data processing, along with links to access all models, is available in our GitHub repository, ensuring transparency and reproducibility of our research.

Evaluation metrics For quantitative assessment of the explanation methods, we followed the evaluation protocol from recent works (Enguehard, 2023; Ferrando et al., 2022) and report results for the following set of metrics: Log-Odds (**LO**) (Shrikumar et al., 2017), Sufficiency (**Suff**), Comprehensiveness (**Comp**) and Area Over the Perturbation Curve (AOPC) for Sufficiency (**A-S**) and Comprehensiveness (**A-C**) (DeYoung et al., 2020). For Suff, LO and A-S the lower the better, while for Comp and A-C the higher the better. Unless specified otherwise, we use the same metric settings as detailed in the referenced papers. A detailed description of the metrics appears in the Appendix (Sec. A.3).

Explanation methods Our evaluation encompasses 9 explanation methods, representing various approaches in the landscape of model explainability. These methods include: GradientXInput (**GXI**) (Shrikumar et al., 2016), GradientShap (**SHAP**) (Lundberg and Lee, 2017), **LIME** (Ribeiro et al., 2016), DeepLift (**LIFT**) (Shrikumar et al., 2017), Integrated Gradients (**IG**) (Sundararajan et al., 2017), Sequential Integrated Gradients (**SIG**) (Enguehard, 2023), GlobEnc (**GLOB**) (Modarressi et al., 2022), **ALTI** (Ferrando et al., 2022), and DecompX (**DCMP**) (Modarressi et al., 2023). The evaluation adhered to the codebase and hyperparameter search settings provided by the original works for each respective method.

Furthermore, we introduce an explanation method based on the LLM Instruct version (**LLM**). In this approach, we prompt the finetuned, open-source versions of Llama-Instruct and Mistral-Instruct with the task and input, instructing the models to explain the prediction by ranking the importance of each token in the input example. Detailed prompts can be found in our GitHub repository for transparency.

Finally, our RPI method was executed with $R = 24$ and $n = 30$. It is worth noting that the results were observed to be robust as long as $n > 30$. Additionally, the computation of attribution maps $\mathbf{m}^{lr}$ (Eq. 4) utilizes the first row of the attention matrices in the case of BERT, DistilBERT and RoBERTa, and the last row of the attention matrices in Llama2 and Mistral. For the exact implementation details, the reader is referred to our

Metric	Prediction	Attribution
Suff	Negative	a hideous , **<u>confusing</u>** spectacle , **one** that may well put the nail in the coffin of any future **rice** adaptations.
	Positive	a **<u>well-made</u>** and often **lovely** depiction of the **mysteries** of friendship
Comp	Negative	a **<u>hideous</u>** , **confusing** spectacle , one that may well put the **nail** in the coffin of any future rice adaptations.
	Positive	a **well-made** and often **<u>lovely</u>** depiction of the mysteries of **friendship**

Table 1: Examples of RPI attributions on several sentences of the SST2 dataset (using the BERT model). The underlined bold words represent the most important tokens in the sentence, while bold words are based on the most important tokens in the sentence, according to the RPI attribution method. The first two and last two rows correspond to the RPI attribution maps that produced the best results for the Suff and Comp metrics, respectively.

GitHub repository.

4.2 Results

We commence with an illustrative example demonstrating that different metrics favor different attributions. Table 1 presents two examples from the SST2 dataset. The first two rows and last two rows visualizes the RPI attribution maps that yielded the best results for the Suff and Comp metrics, respectively, using BERT. In each example, we boldface the top 3 attributed words (according to the generated attribution map), with the word assigned the highest attribution score further underlined. As observed, the best attribution map for each metric differs for the same example. Table 1 illustrates the rationale behind the RPI mechanism, allowing the selection of the most suitable attribution map for each metric. Table 2 presents results for all combinations of encoder-based model, explanation method, dataset, and metric. The findings in Tab 2 consistently highlight the superior performance of RPI across all scenarios, with DCMP and SIG alternating for the second place.

Table 3 illustrates the effectiveness of RPI compared to runner-ups SIG and DCMP from Table 2 on the SST2 dataset using BERT and RoBERTa. The top three attributioned words are extracted for each example based on the explanation method. It is evident that RPI produces attributions that align most closely with the predictions. Additional qualitative examples are provided in the Appendix (Sec. A.2).

Table 4 presents quantitative results for the Llama2 and Mistral models. It is important to note that the ALTI and DCMP methods are not compatible with these models and are therefore excluded from comparison. Similar trends to Tab. 2 emerge, with RPI consistently outperforming other methods by a significant margin in all cases. The runner-up is typically SIG. We also observe that the LLM-based explanation method consistently underperforms compared to the leading explanation methods. This underperformance can be attributed to frequent hallucinations, where the model outputs a ranked token list which includes tokens that are not present in the original input. Overall, the results in Tabs. 2 and 4 establish RPI as the new state-of-the-art method.

4.3 Ablation study

In what follows, we present an ablation study for different configuration choices in RPI. Unless specified otherwise, we used $R = 24$, $n = 30$ and the baselines are drawn from a Gaussian diffusion process. The study was conducted on the RTN dataset, utilizing a finetuned BERT model. Table 6 compares various ablated versions of RPI and explores the impact of applying the RPI baseline resampling procedure to other path-integration methods (IG and SIG). First, we evaluate RPI-G, where attention gradients are used without multiplication in the attention scores (omitting the multiplication in $\mathbf{v}lr$ in Eq.4). Second, we assess RPI-IG and RPI-SIG, which incorporate the RPI baseline resampling procedure into IG and SIG methods, respectively. Here, we replace the fixed <MASK> baseline (Enguehard, 2023) with a random baseline drawn from the Gaussian diffusion process (as described in Sec. 3.2.1), while setting $\mathbf{a}_u^l$ to the <MASK> token.

		RoBERTa					DistilBERT					BERT				
		Suff↓	LO↓	Comp↑	A-S↓	A-C↑	Suff↓	LO↓	Comp↑	A-S↓	A-C↑	Suff↓	LO↓	Comp↑	A-S↓	A-C↑
SST2	RPI	**-0.007**	**-1.691**	**0.491**	**0.008**	**0.250**	**-0.010**	**-2.200**	**0.524**	**0.004**	**0.266**	**-0.019**	**-1.743**	**0.549**	**-0.003**	**0.282**
	SIG	0.097	-1.340	0.381	0.066	0.205	0.064	-1.920	0.453	0.051	0.242	0.083	-1.133	0.378	0.063	0.208
	ALTI	0.116	-1.043	0.305	0.070	0.166	0.059	-1.407	0.368	0.047	0.205	0.093	-0.816	0.320	0.064	0.175
	DCMP	0.086	-1.366	0.393	0.057	0.216	0.359	-0.321	0.089	0.201	0.063	0.037	-1.450	0.467	0.030	0.253
	LIFT	0.331	-0.300	0.098	0.189	0.060	0.234	-0.605	0.162	0.140	0.098	0.289	-0.314	0.118	0.168	0.071
	GLOB	0.227	-0.489	0.166	0.136	0.100	0.357	-0.262	0.073	0.199	0.060	0.233	-0.388	0.153	0.140	0.104
	SHAP	0.227	-0.788	0.226	0.135	0.131	0.207	-1.264	0.291	0.122	0.167	0.250	-0.534	0.191	0.145	0.111
	GXI	0.359	-0.245	0.077	0.200	0.049	0.305	-0.415	0.111	0.179	0.066	0.237	-0.401	0.153	0.140	0.094
	IG	0.116	-1.249	0.360	0.073	0.200	0.100	-1.823	0.415	0.065	0.229	0.110	-0.936	0.334	0.075	0.187
	LIME	0.050	-1.180	0.356	0.035	0.194	0.150	-1.281	0.298	0.104	0.162	0.134	-0.713	0.273	0.093	0.149
RTN	RPI	**-0.029**	**-0.961**	**0.449**	**0.035**	**0.218**	**-0.034**	**-0.559**	**0.390**	**0.033**	**0.186**	**-0.019**	**-2.675**	**0.622**	**0.002**	**0.319**
	SIG	0.114	-0.752	0.324	0.088	0.178	0.087	-0.520	0.316	0.065	0.169	0.157	-1.524	0.353	0.109	0.190
	ALTI	0.146	-0.489	0.228	0.103	0.131	0.119	-0.428	0.206	0.084	0.117	0.111	-1.266	0.334	0.079	0.193
	DCMP	0.072	-0.695	0.314	0.062	0.181	0.294	-0.291	0.068	0.169	0.045	0.045	-2.058	0.471	0.042	0.263
	LIFT	0.374	-0.142	0.059	0.209	0.040	0.232	-0.322	0.095	0.145	0.060	0.310	-0.523	0.155	0.183	0.092
	GLOB	0.229	-0.277	0.139	0.138	0.088	0.309	-0.273	0.046	0.174	0.035	0.269	-0.589	0.179	0.170	0.112
	SHAP	0.218	-0.418	0.188	0.139	0.111	0.171	-0.431	0.213	0.105	0.123	0.310	-0.744	0.181	0.185	0.111
	GXI	0.384	-0.096	0.051	0.215	0.034	0.290	-0.283	0.060	0.172	0.037	0.265	-0.652	0.173	0.163	0.102
	IG	0.121	-0.700	0.316	0.092	0.177	0.095	-0.523	0.322	0.068	0.173	0.183	-1.117	0.278	0.123	0.169
	LIME	0.117	-0.471	0.242	0.086	0.137	0.099	-0.469	0.269	0.072	0.145	0.186	-1.023	0.261	0.124	0.146
AGN	RPI	**-0.015**	-0.814	0.197	0.026	0.117	**-0.008**	**-1.480**	**0.266**	**0.007**	**0.150**	**-0.006**	-1.357	0.231	**0.009**	0.146
	SIG	0.066	-0.664	0.173	0.105	0.110	0.088	-1.089	0.179	0.111	0.107	0.041	-1.422	0.265	0.070	0.163
	ALTI	0.058	-0.724	0.171	0.080	0.126	0.048	-1.076	0.191	0.063	0.143	0.039	-1.443	0.273	0.059	0.192
	DCMP	0.008	**-1.762**	**0.388**	**0.023**	**0.243**	0.300	-0.216	0.043	0.232	0.037	0.002	**-2.656**	**0.445**	0.027	**0.268**
	LIFT	0.221	-0.186	0.047	0.205	0.036	0.202	-0.327	0.062	0.175	0.039	0.308	-0.592	0.110	0.241	0.067
	GLOB	0.110	-0.356	0.086	0.120	0.065	0.167	-0.285	0.054	0.205	0.047	0.074	-0.824	0.161	0.087	0.126
	SHAP	0.201	-0.418	0.098	0.174	0.068	0.212	-0.710	0.115	0.166	0.079	0.223	-0.927	0.159	0.180	0.111
	GXI	0.347	-0.079	0.014	0.268	0.013	0.233	-0.229	0.042	0.208	0.028	0.085	-1.143	0.222	0.101	0.141
	IG	0.088	-0.543	0.144	0.090	0.100	0.085	-1.179	0.190	0.070	0.129	0.060	-1.537	0.264	0.069	0.177
	LIME	0.105	-0.483	0.119	0.114	0.075	0.096	-0.527	0.096	0.119	0.062	0.136	-0.347	0.066	0.147	0.044
EMR	RPI	**0.163**	**-4.449**	**0.668**	0.133	0.350	**0.007**	**-2.400**	**0.683**	**0.017**	**0.359**	**-0.003**	**-4.151**	**0.764**	**0.010**	**0.408**
	SIG	0.347	-2.006	0.438	0.214	0.239	0.147	-1.398	0.494	0.091	0.269	0.190	-2.063	0.512	0.116	0.279
	ALTI	0.172	-3.144	0.608	**0.129**	0.338	0.051	-1.635	0.562	0.036	0.307	0.044	-2.592	0.631	0.030	0.350
	DCMP	0.172	-3.326	0.622	**0.129**	0.337	0.537	-0.370	0.118	0.278	0.082	0.060	-2.763	0.625	0.044	0.346
	LIFT	0.514	-2.020	0.332	0.282	0.183	0.362	-0.891	0.287	0.198	0.157	0.380	-1.422	0.311	0.207	0.174
	GLOB	0.211	-3.106	0.581	0.146	0.322	0.543	-0.369	0.120	0.278	0.084	0.094	-2.423	0.583	0.061	0.322
	SHAP	0.525	-1.293	0.254	0.288	0.151	0.307	-1.024	0.337	0.169	0.186	0.426	-1.157	0.278	0.228	0.167
	GXI	0.541	-1.510	0.258	0.294	0.147	0.415	-0.754	0.237	0.233	0.127	0.354	-1.472	0.330	0.197	0.187
	IG	0.328	-1.917	0.431	0.205	0.237	0.139	-1.396	0.498	0.086	0.269	0.190	-2.106	0.523	0.113	0.285
	LIME	0.198	-3.814	0.647	0.143	**0.351**	0.059	-1.911	0.593	0.041	0.320	0.104	-2.897	0.586	0.065	0.322

Table 2: Evaluation results for all combinations of encoder-based model, dataset, explanation method, and metric. See the notations in Sec. 4.

The results in Tab. 6 indicate that each component in RPI plays a vital yet complementary role, with the baseline resampling procedure showing potential to enhance other path-integration methods. First, RPI-G outperforms RPI-IG, indicating that integration on attention gradients is preferable to integration on the input (as done in IG). Secondly, RPI outperforms RPI-G, demonstrating the benefit of multiplying attention gradients in the attention scores, validating the specific construction of the attribution map in Eq.4. Third, RPI-SIG and RPI-IG outperform SIG and IG, respectively, showcasing the effectiveness of incorporating the RPI baseline resampling procedure. However, both RPI-SIG and RPI-IG fall short compared to RPI, emphasizing the effectiveness of our proposed RPI and the complementary contribution of the RPI baseline resampling procedure along with the attention level integration and the attention scores multiplication. Overall, these findings underscore the efficacy of the RPI baseline resampling procedure as a meta-method to improve performance when applied on top of path-integration methods, and the superiority of the specific RPI implementation pro-

Dataset	Prediction	Model	Method	Attribution
SST2	Positive	BERT	RPI	a modest **pleasure** that **accomplishes** its goals with ease and **confidence** .
			SIG	**a** modest pleasure **that** accomplishes its goals with ease and **confidence** .
			DCMP	a modest **pleasure that** accomplishes its goals with ease and **confidence** .
SST2	Negative	BERT	RPI	apallingly **absurd** ... the chemistry or **lack** thereof between newton and wahlberg **could** turn an imax theater into a 9 " black and white portable tv .
			SIG	apallingly **absurd** ... the chemistry or lack thereof **between** newton and wahlberg could turn an imax theater into a 9 " black and white portable **tv** .
			DCMP	apallingly **absurd** ... the chemistry **or** lack thereof between newton and wahlberg could turn an imax theater into a 9 " black and white portable **tv** .
SST2	Positive	RoBERTa	RPI	not ' **terrible** filmmaking ' **bad** , but more like , ' i once had a nightmare like this , and it 's now coming true ' **bad** .
			SIG	not ' terrible **filmmaking** ' **bad** , but more like , ' i once had a nightmare like this , and it 's now **coming** true ' bad .
			DCMP	not ' terrible **filmmaking** ' **bad** , but more **like** , ' i once had a nightmare like this , and it 's now coming true ' bad .

Table 3: Examples of attributions on sentences from the SST2 dataset. The bold words represent the top 3 words in the sentence, according to each attribution method.

		Llama2					Mistral				
		Suff↓	**LO↓**	**Comp↑**	**A-S↓**	**A-C↑**	**Suff↓**	**LO↓**	**Comp↑**	**A-S↓**	**A-C↑**
SST2	RPI	0.137	**-0.104**	**0.069**	0.083	0.041	**0.205**	**-0.299**	**0.208**	**0.139**	**0.105**
	SIG	0.157	-0.098	0.065	0.090	0.041	0.281	-0.256	0.176	0.160	0.100
	LIFT	0.159	-0.056	0.033	0.093	0.023	0.319	-0.139	0.100	0.184	0.067
	SHAP	0.152	-0.072	0.048	0.088	0.032	0.308	-0.125	0.090	0.177	0.061
	GXI	0.159	-0.054	0.033	0.092	0.023	0.319	-0.142	0.102	0.184	0.067
	LLM	**0.119**	-0.101	0.069	**0.069**	**0.042**	0.257	-0.115	0.082	0.151	0.062
RTN	RPI	0.177	**-0.176**	**0.103**	0.112	**0.061**	**0.189**	**-0.458**	**0.275**	**0.150**	**0.138**
	SIG	0.219	-0.146	0.088	0.126	0.054	0.312	-0.323	0.215	0.184	0.121
	LIFT	0.211	-0.115	0.068	0.124	0.040	0.375	-0.156	0.103	0.220	0.077
	SHAP	0.216	-0.103	0.059	0.124	0.043	0.365	-0.140	0.097	0.213	0.068
	GXI	0.212	-0.114	0.066	0.125	0.040	0.375	-0.161	0.105	0.220	0.077
	LLM	**0.172**	-0.152	0.091	**0.104**	0.056	0.323	-0.129	0.083	0.191	0.065
AGN	RPI	**0.008**	**-1.375**	**0.242**	**0.115**	**0.142**	**-0.008**	**-2.384**	**0.305**	**0.069**	**0.185**
	SIG	0.386	-0.367	0.081	0.298	0.054	0.365	-0.750	0.123	0.295	0.077
	LIFT	0.414	-0.561	0.059	0.312	0.041	0.447	-0.487	0.070	0.334	0.048
	SHAP	0.361	-0.447	0.055	0.291	0.041	0.401	-0.396	0.063	0.311	0.046
	GXI	0.423	-0.564	0.058	0.314	0.040	0.447	-0.489	0.067	0.335	0.048
	LLM	0.252	-0.643	0.047	0.246	0.039	0.359	-0.226	0.042	0.295	0.040
EMR	RPI	**0.259**	**-4.272**	**0.803**	**0.195**	**0.425**	**0.130**	**-5.679**	**0.766**	**0.117**	**0.396**
	SIG	0.469	-1.879	0.584	0.272	0.302	0.273	-2.572	0.620	0.174	0.336
	LIFT	0.661	-1.563	0.456	0.370	0.252	0.629	-1.697	0.367	0.344	0.204
	SHAP	0.632	-1.316	0.386	0.344	0.216	0.451	-2.001	0.452	0.255	0.251
	GXI	0.663	-1.566	0.457	0.369	0.252	0.629	-1.683	0.367	0.344	0.205
	LLM	0.701	-0.773	0.204	0.364	0.124	0.561	-1.306	0.303	0.281	0.185

Table 4: Evaluation results on the Llama2 and Mistral models.

R	Suff↓	LO↓	Comp↑	A-S↓	A-C↑
R=1	0.254	-0.885	0.239	0.154	0.135
R=2	0.097	-1.270	0.342	0.078	0.181
R=4	0.020	-1.725	0.447	0.037	0.233
R=8	-0.008	-2.232	0.534	0.017	0.277
R=16	-0.016	-2.487	0.600	0.007	0.304
R=24	-0.019	-2.675	0.622	0.002	0.319
R=32	**-0.019**	**-2.820**	**0.636**	**0.000**	**0.329**

Table 5: Ablation Study: results for different values of R (the number of sampled baselines in RPI) on RTN using the BERT model.

		Suff↓	LO↓	Comp↑	A-S↓	A-C↑
RTN	RPI	**-0.019**	-2.675	**0.622**	**0.002**	0.319
	RPI-G	0.000	-2.642	0.584	0.017	0.319
	RPI-SIG	-0.013	-2.584	0.600	0.011	0.323
	RPI-IG	-0.017	**-2.703**	0.614	0.004	**0.329**
	IG	0.183	-1.117	0.278	0.123	0.169
	SIG	0.157	-1.524	0.353	0.109	0.190

Table 6: This ablation study highlights significant and complementary contributions of the components in the RPI method. These include integrating at the attention level (RPI) vs. input level (RPI-IG), the benefit from applying the RPI baseline resampling procedure for other path-integration methods (RPI-SIG vs. SIG, RPI-IG vs. IG), and from multiplying the attention scores in their gradients vs. using the plain gradients (RPI vs. RPI-G). The results show that each component in the our proposed RPI method plays a vital role, with the baseline resampling procedure showing potential to enhance other path-integration methods as well.

posed in this work. Table 5 investigates the impact of varying the number of sampled baselines for $R \in \{1, 2, 4, 8, 16, 24, 32\}$ on RPI's performance. We observe a slight improvement when increasing the number of trials from 24 to 32, demonstrating that the settings of $R = 24$ are sufficient for achieving state-of-the-art performance while maintaining acceptable runtime. It is noteworthy that although the number of interpolation steps n is also a configurable parameter, increasing it beyond $n = 30$ did not result in a significant improvement.

Lastly, recall that in the general case, RPI supports the inclusion of attribution maps from all layers in the model (Eq. 5). Therefore, in Appendix A.1, we provide a thorough ablation study assessing the performance of RPI when applied to individual layers and when aggregating attribution maps from multiple layers. Our findings show that performance tends to improve with deeper layers, and top-down layer aggregation performs the best.

5 Conclusion

This work responds to the diversity inherent in explanation metrics, recognizing that different metrics may promote different attribution maps. RPI effectively addresses this challenge by introducing randomness into the integration path through random baseline sampling, thereby generating a pool of candidate attribution maps. The adaptability provided by multiple baselines enables RPI to select the most effective attribution map tailored to the specific evaluation metric. Through an extensive evaluation encompassing 11 explanation methods, 5 language models, and 4 datasets, our work establishes the superiority of RPI over current state-of-the-art methods across a diverse range of explanation metrics. These findings highlight the effectiveness of RPI as a machinery for explaining LMs.

6 Limitations and Future Work

While our RPI method has demonstrated its effectiveness in providing state-of-the-art results by sampling baselines from a Gaussian diffusion process, there exist certain limitations that merit consideration and avenues for future research.

Firstly, to enhance the variety of drawn baselines and, consequently, the resulting attribution maps, an exploration of additional baseline distributions is warranted. One potential avenue is to model the baseline distribution $\mathcal{B}$ using a more diverse distribution, such as a mixture of distributions, as suggested in Section 3.

Another limitation lies in the non-adaptive nature of the sampling process in RPI. Adaptive sampling techniques can leverage information from already drawn baselines to intelligently decide the next region in space from which to draw baselines, thereby potentially improving performance on the metric of interest. Exploration-exploitation approaches, updating the baseline distribution in an online manner as sampling process evolves, could be explored to address this limitation.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

Arthur Mensch Chris Bamford Devendra Singh Chaplot Diego de las Casas Florian Bressand Gianna Lengyel Guillaume Lample Lucile Saulnier Lélio Renard Lavaud Marie-Anne Lachaux Pierre Stock Teven Le Scao Thibaut Lavril Thomas Wang Timothée Lacroix William El Sayed Albert Q. Jiang, Alexandre Sablayrolles. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629.

Oren Barkan. 2017. Bayesian neural word embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Oren Barkan, Omri Armstrong, Amir Hertz, Avi Caciularu, Ori Katz, Itzik Malkiel, and Noam Koenigstein. 2021a. Gam: Explainable visual similarity and classification via gradient activation maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 68–77.

Oren Barkan, Yuval Asher, Amit Eshel, Yehonatan Elisha, and Noam Koenigstein. 2023a. Learning to explain: A model-agnostic framework for explaining black box models. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 944–949. IEEE.

Oren Barkan, Veronika Bogina, Liya Gurevitch, Yuval Asher, and Noam Koenigstein. 2024. A counterfactual framework for learning and evaluating explanations for recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pages 3723–3733.

Oren Barkan, Avi Caciularu, Ori Katz, and Noam Koenigstein. 2020a. Attentive item2vec: Neural attentive user representations. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3377–3381. IEEE.

Oren Barkan, Avi Caciularu, Idan Rejwan, Ori Katz, Jonathan Weill, Itzik Malkiel, and Noam Koenigstein. 2020b. Cold item recommendations via hierarchical item2vec. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 912–917. IEEE.

Oren Barkan, Avi Caciularu, Idan Rejwan, Ori Katz, Jonathan Weill, Itzik Malkiel, and Noam Koenigstein. 2021b. Representation learning via variational bayesian networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 78–88.

Oren Barkan, Yehonatan Elisha, Yuval Asher, Amit Eshel, and Noam Koenigstein. 2023b. Visual explanations via iterated integrated attributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2073–2084.

Oren Barkan, Yehonatan Elisha, Jonathan Weill, Yuval Asher, Amit Eshel, and Noam Koenigstein. 2023c. Deep integrated explanations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 57–67.

Oren Barkan, Yehonatan Elisha, Jonathan Weill, Yuval Asher, Amit Eshel, and Noam Koenigstein. 2023d. Stochastic integrated explanations for vision models. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE.

Oren Barkan, Yonatan Fuchs, Avi Caciularu, and Noam Koenigstein. 2020c. Explainable recommendations via attentive multi-persona collaborative filtering. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 468–473.

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. 2021c. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2882–2887.

Oren Barkan, Roy Hirsch, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2021d. Anchor-based collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2877–2881.

Oren Barkan, Ori Katz, and Noam Koenigstein. 2020d. Neural attentive multiview machines. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3357–3361. IEEE.

Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019a. Cb2cf: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 228–236.

Oren Barkan, Noam Razin, Itzik Malkiel, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2020e. Scalable attentive sentence pair modeling via distilled sentence embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3235–3242.

Oren Barkan, Tal Reiss, Jonathan Weill, Ori Katz, Roy Hirsch, Itzik Malkiel, and Noam Koenigstein. 2023e.

Efficient discovery and effective evaluation of visual perceptual similarity: A benchmark and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20007–20018.

Oren Barkan, Idan Rejwan, Avi Caciularu, and Noam Koenigstein. 2020f. Bayesian hierarchical words representation learning. In *"Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics"*, pages 3871–3877.

Oren Barkan, Tom Shaked, Yonatan Fuchs, and Noam Koenigstein. 2023f. Modeling users' heterogeneous taste with diversified attentive user profiles. *User Modeling and User-Adapted Interaction*, pages 1–31.

Oren Barkan, Shlomi Shvartzman, Noy Uzrad, Almog Elharar, Moshe Laufer, and Noam Koenigstein. 2023g. Inversynth ii: Sound matching via self-supervised synthesizer-proxy and inference-time finetuning. ISMIR.

Oren Barkan and David Tsiris. 2019. Deep synthesizer parameter estimation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3887–3891. IEEE.

Oren Barkan, David Tsiris, Ori Katz, and Noam Koenigstein. 2019b. Inversynth: Deep estimation of synthesizer parameter configurations from audio signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2385–2396.

Shay Ben-Elazar, Gal Lavee, Noam Koenigstein, Oren Barkan, Hilik Berezin, Ulrich Paquet, and Tal Zaccai. 2017. Groove radio: A bayesian hierarchical model for personalized playlist generation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 445–453.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660.

Hila Chefer, Shir Gur, and Lior Wolf. 2021a. Generic attention-model explainability for interpreting bimodal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406.

Hila Chefer, Shir Gur, and Lior Wolf. 2021b. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. 2020. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*.

Joseph Enguehard. 2023. Sequential integrated gradients: a simple but effective method for explaining language models. *arXiv preprint arXiv:2305.15853*.

Javier Ferrando, Gerard I Gállego, and Marta R Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. *arXiv preprint arXiv:2203.04212*.

Ruth Fong, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958.

Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437.

Keren Gaiger, Oren Barkan, Shir Tsipory-Samuel, and Noam Koenigstein. 2023. Not all memories created equal: Dynamic user representations for collaborative filtering. *IEEE Access*, 11:34746–34763.

Dvir Ginzburg, Itzik Malkiel, Oren Barkan, Avi Caciularu, and Noam Koenigstein. 2021. Self-supervised document similarity ranking via contextualized language models and hierarchical inference. *arXiv preprint arXiv:2106.01186*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Ori Katz, Oren Barkan, Noam Koenigstein, and Nir Zabari. 2022. Learning to ride a buy-cycle: A hyper-convolutional model for next basket repurchase recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 316–326.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976.

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

Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, and Noam Koenigstein. 2020. Recobert: A catalog language model for text-based recommendations. *arXiv preprint arXiv:2009.13292*.

Itzik Malkiel, Dvir Ginzburg, Oren Barkan, Avi Caciularu, Jonathan Weill, and Noam Koenigstein. 2022a. Interpreting bert-based text similarity via activation and saliency maps. In *Proceedings of the ACM Web Conference 2022*, pages 3259–3268.

Itzik Malkiel, Dvir Ginzburg, Oren Barkan, Avi Caciularu, Yoni Weill, and Noam Koenigstein. 2022b. Metricbert: Text representation learning via self-supervised triplet training. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *NAACL-HLT*.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. Decompx: Explaining transformers decisions by propagating token decomposition. *arXiv preprint arXiv:2306.02873*.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. Globenc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. *arXiv preprint arXiv:2205.03286*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. *arXiv preprint arXiv:2108.13654*.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. FaiRR: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093, Dublin, Ireland. Association for Computational Linguistics.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2020. Visualizing the impact of feature attribution baselines. *Distill*. Https://distill.pub/2020/attribution-baselines.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. 2018. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*.

A Appendix

A.1 Layer ablation study

The evaluation results in Sec. 4.2 indicate that applying RPI solely to the last layer consistently yields state-of-the-art results. While this approach aligns with previous seminal explanation methods (Selvaraju et al., 2017; Caron et al., 2021) that predominantly utilize the final layer of the model for explanation generation, the general formulation of RPI, as described in Eq. 5, allows for the utilization of any arbitrary layer in the model and their combinations. Therefore, for the sake of completeness, in this section, we present an ablation study investigating the performance of RPI when applied to each individual layer in the model separately, as well as the benefits of aggregating attribution maps from multiple layers in the model.

To this end, we ablate over the layers in three different ways:

1. Individual Layers: We assess RPI's performance when applied to each layer separately. This is achieved by setting J (Eq. 5) to a set containing the individual layer of interest each time.

2. Top-down Aggregation: RPI's performance is evaluated when applied to a set of consecutive layers in a top-down manner. This involves combining the attribution maps produced starting from the top layer L (the last layer) downwards to layer $L-i$, with $i \in \{0, 1, ..., 9\}$. Specifically, this is obtained by setting $J = \{L, L-1, ..., L-i\}$ in Eq. 5.

3. Bottom-up Aggregation: We assess RPI's performance in a bottom-up manner, starting from the first layer and up to layer $L-i$. This is achieved by setting $J = 1, 2, ..., L-i$ in Eq. 5.

Table 7 presents the results obtained based on the RTN dataset using an RTN finetuned BERT model (L stands for the last layer in the model). The Individual Layers section presents RPI results when applied to each individual layer $L-i$ separately, for $i \in \{0, 1, ..., 11\}$. The Top-down Aggregation section presents RPI results for the combination of attribution maps produced by multiple consecutive layers in a top-down manner. Specifically, the row associated with the layer $L-i$ corresponds to the application of Eq. 5 with $J = \{L, L-1, ..., L-i\}$. Conversely, in the Bottom-up Aggregation section, RPI is applied with $J = \{1, 2, ..., L-i\}$ for the layer $L-i$.

It is important to clarify that the results in the 'Top-down Aggregation' and 'Bottom-up Aggregation' sections are derived from aggregating the attribution maps produced from each individual layer. Specifically, the RPI procedure was applied **once** for each individual layer, and the resulting attribution maps were then used for both top-down and bottom-up aggregations. Consequently, the results for full aggregation across all layers are identical for both top-down and bottom-up approaches (found in the first and last rows of the 'Top-down Aggregation' and 'Bottom-up Aggregation' sections, respectively). However, in all other cases, top-down aggregation outperforms bottom-up aggregation. For instance, aggregating the attribution maps from last 3 layers (corresponding to the row for $L-2$ in the 'Top-down Aggregation' section) demonstrates superior performance compared to the aggregation of the attribution maps from the first 3 layers (corresponding to the row for $L-7$ in the 'Bottom-up Aggregation' section).

Overall, the results in Tab. 7 indicate that performance tends to improve with deeper layers when applied individually, and as a result, it is preferred to aggregate attribution maps from the layers in a top-down manner.

A.2 Additional qualitative results

Table 8 showcases various RPI attributions obtained from multiple instances in the SST2 and EMR datasets, employing the RoBERTa model. The 'Prediction' column denotes the class with the highest prediction score by the model. In each example, the top three ranked words according to RPI attribution are highlighted in bold. Notably, the highlighted words exhibit semantic similarity to the predicted class, thereby offering a plausible explanation that supports the model's prediction.

A.3 Evaluation metrics

For quantitative assessment of the explanation methods, we consider the following set of metrics:

1. Log-Odds (**LO**) (Shrikumar et al., 2017) score is defined as the average difference of the

	Individual Layers					Bottom-up Aggregation					Top-down Aggregation				
	Suff↓	**LO↓**	**Comp↑**	**A-S↓**	**A-C↑**	**Suff↓**	**LO↓**	**Comp↑**	**A-S↓**	**A-C↑**	**Suff↓**	**LO↓**	**Comp↑**	**A-S↓**	**A-C↑**
L	-0.018	-2.583	0.610	0.005	0.309	**-0.021**	**-3.357**	**0.725**	**-0.010**	**0.369**	-0.018	-2.583	0.610	0.005	0.309
$L-1$	**-0.019**	-2.573	0.609	0.003	0.314	-0.021	-3.301	0.718	-0.010	0.367	-0.019	-2.909	0.657	-0.002	0.335
$L-2$	-0.019	**-2.675**	**0.622**	**0.002**	**0.319**	-0.021	-3.233	0.713	-0.009	0.363	-0.020	-3.108	0.686	-0.004	0.350
$L-3$	-0.018	-2.662	0.617	0.003	0.315	-0.021	-3.113	0.700	-0.009	0.355	-0.020	-3.221	0.699	-0.006	0.358
$L-4$	-0.018	-2.306	0.574	0.010	0.284	-0.021	-2.945	0.681	-0.009	0.344	-0.021	-3.263	0.708	-0.007	0.361
$L-5$	-0.018	-2.138	0.538	0.010	0.274	-0.021	-2.816	0.658	-0.008	0.336	-0.021	-3.290	0.710	-0.008	0.363
$L-6$	-0.018	-2.093	0.537	0.004	0.278	-0.021	-2.700	0.645	-0.007	0.329	-0.021	-3.313	0.714	-0.008	0.365
$L-7$	-0.018	-1.881	0.500	0.009	0.253	-0.021	-2.555	0.621	-0.006	0.317	-0.021	-3.324	0.718	-0.009	0.366
$L-8$	-0.018	-1.919	0.516	0.009	0.261	-0.020	-2.444	0.600	-0.004	0.309	-0.021	-3.336	0.722	-0.009	0.367
$L-9$	-0.017	-1.814	0.479	0.008	0.254	-0.020	-2.216	0.560	-0.001	0.290	-0.021	-3.347	0.723	-0.009	0.368
$L-10$	-0.017	-1.615	0.447	0.023	0.237	-0.019	-1.887	0.508	0.014	0.263	-0.021	-3.352	0.725	-0.010	0.369
$L-11$	-0.013	-1.421	0.411	0.047	0.216	-0.013	-1.421	0.411	0.047	0.216	**-0.021**	**-3.357**	**0.725**	**-0.010**	**0.369**

Table 7: RPI layer ablation study using a finetuned version of BERT on the RTN dataset. L stands for the last layer in the model. The Individual Layers section presents RPI results when applied each individual layer in the model separately. This is achieved by setting J (Eq. 5) to a set containing the individual layer of interest each time. The Top-down Aggregation section presents RPI results when applied to multiple consecutive layers starting from the top layer L (the last layer) downwards. Specifically, the row associated with the layer $L-i$ corresponds to the application of RPI with $J=\{L, L-1, ..., L-i\}$. Conversely, in the Bottom-up Aggregation section, RPI is applied with $J=\{1, 2, ..., L-i\}$ for the layer $L-i$. The results indicate that performance tends to improve with deeper layers when applied to a single layer, and therefore, it is preferred to aggregate attribution maps from the layers in a top-down manner.

Dataset	**Prediction**	**Attribution**
EMR	Joy	i continue to **feel** so **content** about our **decision** to move here
	Fear	i have to take jenny in to be spayed so of course im **feeling** **nervous** and **guilty**
	Sadness	im **feeling** sentimental or in **need** of **reassurance**
	Fear	i was **feeling** a **little** **fearful** of trying to eat this damn thing
	Anger	i remember **feeling** so **hellip** **furious** with the shooter
	Anger	i just **feel** really **violent** right **now**
	Joy	i feel so **tranquil** right now **its** **great**
	Fear	i couldn t turn my head away even **when** **i** feel **frightened**
	Surprise	**i** **feel** so deeply **shocked** and saddened
	Fear	**im** **feeling** a combination of **terrified** and relieved
SST2	Negative	first , for a movie **that** tries to be **smart** , it 's kinda **dumb**
	Negative	**tedious** norwegian offering which **somehow** snagged an oscar **nomination**
	Positive	the film retains **ambig**uities **that** make it well **worth** watching
	Negative	makes piecing **the** story together **frustrating** **difficult**
	Negative	doubtful this list**less** feature will **win** him any new viewers
	Positive	to **emerge** as an **exquisite** **motion** picture in its own right
	Positive	one of the most **interesting** **writer**/direct**ors** working today
	Positive	would be forgettable if it were n't such **a** **clever** **adaptation** of the bard 's tragic play

Table 8: Examples of RPI attributions for multiple examples from the SST2 and EMR datasets (utilizing the RoBERTa model). The 'Prediction' column indicates the class associated with the highest prediction score by the model. In each example, the top three ranked words according to RPI attribution are marked in bold. The highlighted words demonstrate semantic similarity to the predicted class, thereby providing a reasonable explanation that supports the model's prediction.

negative logarithmic probabilities on the predicted class before and after masking the top $k\%$ words with <MASK> padding (Encoder-based models) and <UNK> padding (Llama2). Lower scores are better. In this work, we used LO with $k=20$.

2. Comprehensiveness (**Comp**) (DeYoung et al., 2020) score is defined as the average difference of the change in predicted class probability before and after removing the top $k\%$ features. Similar to Log-odds, this measures the influence of the top-attributed tokens on the model's prediction. For a single example Comp is computed as:

$$p(y'|x_i) - p(y'|x_i^{(k)}),$$

where y' is the predicted class, x is the input sequence of tokens, and $x^{(k)}$ denotes the modified sequence with the top $k\%$ attributed tokens deleted from the sequence. Higher scores are better. In this work, we used Comp with $k = 20$.

3. Sufficiency (**Suff**) (DeYoung et al., 2020) score is defined as the average difference of the change in predicted class probability before and after keeping only the top $k\%$ tokens. This measures the adequacy of the top $k\%$ attributions for model's prediction. It is defined in a similar fashion as comprehensiveness, except the $x^{(k)}$ is defined as the sequence containing only the top $k\%$ tokens. Lower scores are better. In this work, we used Suff with $k = 20$.

4. Area Over the Perturbation Curves (AOPC): AOPC-Sufficiency (**A-S**) and AOPC-Comprehensiveness (**A-C**) (DeYoung et al., 2020) - are the average differences of the change in predicted class probability before and after keeping and removing the top $k\%$ tokens for Sufficiency and Comprehensiveness, respectively:

$$\text{AOPC-S} = \frac{1}{|B|} \sum_{k \in B} \text{Suff}(k),$$

$$\text{AOPC-C} = \frac{1}{|B|} \sum_{k \in B} \text{Comp}(k).$$

Here, we evaluate Comp and Suff for 5 different values of k, setting $B = \{1, 5, 10, 20, 50\}$ as suggested by (DeYoung et al., 2020). A-S and A-C measure how well a specific token ordering is scored under a model from two complementary perspectives and across the k axis.

A.4 LLM prompts

This section outlines the prompts used by the Llama2 and Mistral models for each dataset in a few-shot context. Utilizing these prompts led to classification accuracy exceeding 90% for the SST2 and RTN tasks, which is also true for the LoRA-based finetuned models for the AGN and EMR tasks. It is noteworthy that in a zero-shot mode, the results experience a significant degradation.

A.4.1 SST2 prompt

Classify the sentiment of sentences. for each sentence the label is positive (P) or negative (N)

Text: hide new secretions from the parental units
Label:N

Text: the greatest musicians
Label:P

Text:are more deeply thought through than in most ‘ right-thinking ’ films
Label:P

Text: on the worst revenge-of-the-nerds clichés the filmmakers could dredge up
Label:0

Text:[text]
Label:

A.4.2 RTN prompt

Classify the sentiment of sentences. for each sentence the label is positive (P) or negative (N)

Text: the film desperately sinks further and further into comedy futility.
Label:N

Text: if you sometimes like to go to the movies to have fun , wasabi is a good place to start
.
Label:P

Text: plays like the old disease-of-the-week small-screen melodramas .

Label:N

Text: hip-hop has a history , and it's a metaphor for this love story .
Label:P

Text: spiderman rocks
Label:P

Text:so exaggerated and broad that it comes off as annoying rather than charming .
Label:N

Text:[text]
Label: