Deep Integrated Explanations

A thesis submitted
in partial fulfilment of the requirements
for the Degree Master of Science in Computer Science
By Yehonatan Elisha

The Open University Of Israel
August 24, 2023

Abstract

This research work presents Deep Integrated Explanations (DIX) - a universal method for explaining vision models. DIX generates explanation maps by integrating information from the intermediate representations of the model, coupled with their corresponding gradients. Notably, our method's unique utilization of multiple layer integration proves instrumental in producing explanation maps that are both faithful and accurate. Through an extensive array of both objective and subjective evaluations spanning diverse tasks, datasets, and model configurations, we showcase the robustness of DIX, surpassing current state-of-the-art methods.

האוניברסיטה הפתוחה
המחלקה למתמטיקה ולמדעי המחשב

הסברים למודלי ויז'ן בעזרת אינטגרציה עמוקה

עבודת תזה זו הוגשה כחלק מהדרישות לקבלת תואר

"מוסמך למדעים" .M.Sc במדעי המחשב

באוניברסיטה הפתוחה

החטיבה למדעי המחשב

על-ידי

יהונתן אלישע

העבודה הוכנה בהדרכתו של ד"ר אורן ברכאן

דצמבר 2023

תקציר

בעבודה זו מוצגת שיטה גנרית ופשוטה בשם Deep Integrated Explanation (DIX) להסברת מודלים לראייה ממוחשבת. שיטה זו יוצרת מפות הסברה בעזרת אינטגרציה על המידע מייצוגי השכבות השונות ברשת בשילוב עם הגרדיאנט שלהם. בוצע ניסוי רחב שכלל מגוון של מדדים בתחום הסברת המודלים, שימוש במגוון גדול של רשתות, דאטה-סטים והשוואה להרבה שיטות שנחשבות ל-state-of-the-art בתחום הסברת המודלים. תוצאות הניסוי מדגיש את אפקטיביות השיטה והיכולת שלה לספק תוצאות טובות יותר משאר השיטות. לאורך תהליך המחקר, התבצעה בחינה מעמיקה של אפשרויות הגדרה שונות ל-DIX בכדי להבין אילו אלמנטים יוכלו להועיל בצורה המיטבית ליצירת המפות המוצלחות ביותר. לבסוף, בוצעו בדיקות שפיות לשיטות הסברה אשר הראו את רגישות השיטה לשינוי במידע המתויג או במשקולות המודל.

Table of Contents

Acknowledgments

First, I would like to thank my supervisor, Dr. Oren Barkan, for the consistent support, patience and investment in both my professional and personal growth throughout the entirety of the research process. Thanks to you I adapted new skills and learned new cognitive perspectives. Furthermore, you increased my enthusiasm for the research world.

Thanks to my research colleagues, Yuval Asher and Amit Eshel, who helped me with data collection and metric selection. Additionally, I extend my gratitude to the HPC team of the department, whose exceptional assistance and unwavering support in resolving computational challenges have been invaluable throughout my research journey.

Finally, special thanks to the academic and administrative staff of the Computer Science department, who were always there to advise and help.

List of Figures

List of Tables

1 Introduction

In the present landscape of computer vision, deep Convolutional Neural Networks (CNNs) [1, 2, 3, 4], alongside recent Vision Transformers (ViTs) models [5, 6] have risen to prominence, exhibiting outstanding performance in a variety of vision tasks [1, 7, 8, 9]. This surge in popularity emphasizes the need to comprehend the underlying rationale driving the decisions and predictions of deep learning models.

Despite their remarkable achievements, most deep neural networks remain enigmatic, often considered black boxes due to their vast number of parameters and intricate non-linearities. This opacity has ignited the growth of explainable AI as a focal research area within the realm of deep learning. Consequently, numerous methodologies have been proposed for explaining the predictions of deep learning models in computer vision [10, 11, 12, 13], natural language processing [14, 15], and recommender systems [16, 17, 18].

Explanation techniques aim to bridge the gap in understanding by generating heatmap-like explanation maps. These maps spotlight distinct input regions, attributing predictions to specific areas within the input image. Initially, rooted in gradient-based approaches, early methods generated explanation maps by analyzing the gradient of predictions concerning the input image [11, 1, 19]. Subsequently, several works [12, 20, 21, 22] proposed deriving explanation maps from the internal activation maps produced by the network, along with their gradients. Other techniques, such as Integrated Gradients (IG) [23], relying on path integration, created explanation maps by accumulating gradients from linear interpolations between input and reference images.

Predominantly applied to CNNs, the aforementioned methods arose before the emergence of Transformer-based architectures [24]. With the advent of ViT models [25], a variety of methodologies were proposed to interpret and explain them, including recent explanation techniques like those presented in [13, 26].

This work introduces Deep Integrated Explanations (DIX), a comprehensive approach aimed at explaining vision models, which finds applicability across both CNN and ViT architectures. DIX employs integration over the internal model representations and their gradients, facilitating the extraction of insights from any activation (or attention) map within the network.

We present a thorough objective and subjective evaluation, showcasing the efficacy of DIX on both CNN and ViT models. Our results reveal its superiority over other baselines

across various explanation and segmentation tasks, encompassing diverse datasets, model architectures, and evaluation metrics. Additionally, we validate the credibility of DIX in producing faithful explanation maps through an extensive set of sanity tests, as outlined in [27].

2 Explainable AI and Interpretable AI

In recent years, the increasing integration of artificial intelligence (AI) systems into various domains has given rise to concerns about their opacity and lack of transparency. This has led to the emergence of two closely related concepts: Explainable AI (XAI) and Interpretable AI (IAI). These concepts aim to enhance our understanding of AI models and their decision-making processes, fostering trust, accountability, and regulatory compliance. Although often used interchangeably, explainable AI and interpretable AI encompass distinct paradigms, each addressing specific challenges associated with AI system opacity. This section delves into the fundamental differences between Explainable AI and Interpretable AI, highlighting their underlying principles, methodologies, and implications.

Explainable AI: A Holistic Understanding Explainable AI, often referred to as XAI, emphasizes the provision of human-understandable explanations for the decisions made by AI models. The objective of XAI is to bridge the gap between the complex inner workings of AI algorithms and the comprehension of non-expert users. The key emphasis here is on the comprehensibility of the explanation rather than the transparency of the model itself. XAI techniques aim to answer questions such as "Why did the AI make this decision?" by providing insights into the features, data points, or reasoning that contributed to a particular outcome. Techniques encompassed within the realm of XAI include:

1. Feature Attribution: This involves identifying the contribution of individual features or input data to the model's output. Techniques like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) fall under this category.

2. Rule-based Explanations: These approaches seek to generate explanations in the form of human-understandable rules, enabling users to comprehend decision logic. Rule-based systems often use decision trees or if-then-else constructs.

3. Example-based Explanations: Explanations are provided by presenting examples similar to the input data that led to a particular decision. This technique can be particularly useful in visual domains like image classification.

Interpretable AI: Unveiling Model Transparency Interpretable AI, or IAI, is concerned with the transparency and comprehensibility of the AI model itself. Unlike XAI, which focuses on post hoc explanations, IAI aims to build models that are inherently understandable. An interpretable AI model is one in which the relationship between inputs and outputs can be directly inferred, facilitating intuitive reasoning. IAI techniques seek to simplify model architectures, reduce complexity, and ensure that the model's decision-making process aligns with human cognitive capabilities. Techniques associated with Interpretable AI encompass:

1. Linear Models: Linear models, such as linear regression and logistic regression, offer inherent interpretability due to their clear relationship between inputs and outputs.

2. Decision Trees: Decision trees partition data based on features, leading to easily interpretable decision paths. However, they might struggle with capturing complex relationships.

3. Sparse Models: These models emphasize the use of a limited number of features, enhancing interpretability by focusing on the most relevant inputs.

4. Symbolic Models: Symbolic AI approaches aim to represent knowledge in a symbolic form, making the reasoning process explicit and human-readable.

Conclusion Explainable AI and Interpretable AI represent two intertwined yet distinct approaches in addressing the challenges of AI transparency and understandability. While Explainable AI concentrates on providing comprehensible explanations for AI decisions, Interpretable AI focuses on constructing models with inherent transparency. The convergence of these paradigms holds the potential to not only improve the adoption of AI systems but also to ensure responsible and accountable AI deployment across various applications. As AI continues to evolve, the ongoing development and integration of both XAI and IAI techniques will play a crucial role in shaping the future of AI technology.

Our research is centered on the domain of Explainable AI (XAI). Within this context, we introduce a simple and versatile approach designed to explain vision models. Our method leverages insights extracted from the intermediary feature representations of the model, coupled with their corresponding gradients. This aggregation of information is employed to generate an output akin to a heatmap, thereby highlighting the pixels that have exerted the most pronounced influence on the model's prediction. This outcome, denoted as an

explanation map, serves as a visual representation of the significant contributors to the model's predictions in the computer vision explainability world.

3 Related Work

3.1 Explanation Methods for CNNs

A diverse range of explanation methods were proposed for explaining CNN models, categorized into various types including perturbation-based methods, gradient methods, saliency-based methods,and gradient-free methods. Perturbation-based methods [28, 29] gauge output sensitivity concerning input through random perturbations applied in the input space. Saliency-based methods [30, 11, 31, 32, 10, 33] leverage feature maps obtained through forward propagation to interpret model predictions.

Gradient methods utilize prediction gradients with respect to the input or intermediate activation maps. These methods yield explanation maps based on the gradient itself or by a combination of the activation maps with their gradients [34, 35]. For instance, SmoothGrad [36] presents a smoothing approach, applied by adding random Gaussian noise to the input image at each iteration. Another notable example is the Grad-CAM (GC) [12] method, which leverages activation maps from the final convolutional layer in conjunction with their pooled gradients to generate explanation maps. The effectiveness of GC has subsequently inspired numerous follow-up work [20, 37, 21, 22].

Gradient-free methods generate explanation maps by manipulating activation maps without relying on gradient information [38, 39]. For instance, LIFT-CAM [40] utilizes the DeepLIFT [41] technique to estimate SHAP values of activation maps [42], which are then combined with the activation maps to produce the explanation map. However, gradient-free methods have a drawback: they neglect gradient information, thereby constraining their ability to steer explanations toward the target or predicted class.

Finally, a notable avenue of research pertains to path integration methods. Integrated Gradients (IG) [23] involves integration across interpolated image gradients. Blur IG (BIG) [43] focuses on introducing information using a baseline and adopts a path that gradually removes Gaussian blur from the attributed image. Guided IG (GIG) [44] refines IG by introducing an adaptive path strategy. By computing integration along an alternative path, it circumvents high gradient regions, often resulting in a reduction of irrelevant attributions.

Distinguished from the aforementioned works, DIX employs integration, facilitating interpolation on the internal representations produced by the network, and offers to combine the resulting explanation maps from all network layers. Furthermore, DIX does not confine

the integrand to simple gradients, but rather encompasses an arbitrary function involving the activation (attention) maps and their gradients.

3.2 Explanation Methods for ViTs

Early attempts to explain Transformers employed the attention scores inherent to ViT models in order to glean insights w.r.t. the input [24, 45]. However, it is not clear how to combine the scores from different layers. Simple averaging the attention scores of each token, for example, leads to blurring of the signal [13].

Abnar and Zuidema [46] proposed the Rollout method to compute attention scores to input tokens at each layer by considering raw attention scores in a layer as well as those from precedent layers. Rollout improved results over the utilization of a single attention layer. However, by relying on simplistic aggregation assumptions, irrelevant tokens often become highlighted. LRP [47], proposed to propagate gradients from the output layer to the beginning, considering all the components in the transformer's layers and not just the attention layers.

Recently, Chefer et al.[13] introduced Transformer Attribution (T-Attr), a class-specific Deep Taylor Decomposition method that employs relevance propagation for both positive and negative attributions. More recently, the same authors introduced Generic Attention Explainability (GAE)[26], which is an extension of T-Attr aimed at explaining Bi-Modal transformers. T-Attr and GAE stand as state-of-the-art methods for explaining ViT models, exhibiting superior performance when compared to other effective explanation methods, including LRP and partial LRP [48].

DIX differs from T-Attr and GAE in two main aspects: First, DIX is a versatile method capable of producing explanation maps for both CNNs and ViTs. Second, in the context of ViT models, DIX employs path integration on the interpolated attention matrices while incorporating the Gradient Rollout (GR) representation (a variant of the Rollout method) as the function for integration.

4 Method

Let $f : \mathbb{R}^{D_0} \to \mathbb{R}^K$ be a neural network with L hidden layers that takes an input (image) $\mathbf{x} \in \mathbb{R}^{D_0}$ and produces a prediction $f(\mathbf{x}) \in \mathbb{R}^K$. We denote $\mathbf{x}^l$ $(1 \leq l \leq L)$ as the intermediate representation generated by the l-th hidden layer in f (based on the input $\mathbf{x}$), and $f^l : \mathbb{R}^{D_l} \to \mathbb{R}^K$ as the sub network of f that takes $\mathbf{x}^l$ as an input and outputs the prediction $f(\mathbf{x})$. Consequently, we have the relationship $f^l(\mathbf{x}^l) = f(\mathbf{x})$. Additionally, we denote $\mathbf{x}^0 = \mathbf{x}$ and $f^0 = f$.

Our assumption is that $\mathbf{x}^l$ preserves the spatial structure of $\mathbf{x}$ (though at a different resolution) such that each element in x^l is associated with its corresponding elements in $\mathbf{x}$ (e.g., this assumption holds true for CNNs). W.l.o.g, we restrict the discussion to multi-class classification problems, hence f outputs a vector assigning score to each class, and the score for the class k is denoted as $f_k(\mathbf{x})$.

Our objective is to *explain* the prediction $f_k(\mathbf{x})$ for the class k. In this work, we define an *explanation map* $\mathbf{m}^l$ as a tensor assigning an attribution score to each element in $\mathbf{x}^l$ w.r.t. the prediction $f_k^l(\mathbf{x}^l) = f_k(\mathbf{x})$. Consequently, $\mathbf{m}^l$ must match the dimensions of $\mathbf{x}^l$. Note that our ultimate goal is to attribute the prediction to each element in the input $\mathbf{x}$, and due to the spatial structure preservation, each element in $\mathbf{m}^l$ can be associated with a set of elements in $\mathbf{x}$.

Let $\mathbf{z}^l \in \mathbb{R}^{D_l}$ be a baseline serving as a reference for the informative representation $\mathbf{x}^l$. $\mathbf{z}^l$ can be the null representation, random noise, or other baselines representing missing information. In what follows, we present a decomposition of the score difference $f_k(\mathbf{x}) - f_k^l(\mathbf{z}^l)$, from which an explanation map $\mathbf{m}^l$ is derived.

Let C^l be a differentiable curve connecting $\mathbf{z}^l$ to $\mathbf{x}^l$. C^l is parameterized by a vector function $\mathbf{r}^l : [0, 1] \to \mathbb{R}^{D_l}$ such that $\mathbf{r}^l(0) = \mathbf{z}^l$ and $\mathbf{r}^l(1) = \mathbf{x}^l$. The score difference $f_k(\mathbf{x}) - f_k^l(\mathbf{z}^l)$ can then be expressed as follows:

$$
\begin{aligned}
f_k(\mathbf{x}) - f_k^l(\mathbf{z}^l) &= f_k^l(\mathbf{r}^l(1)) - f_k^l(\mathbf{r}^l(0)) \\
&= \int_0^1 \frac{d}{dt} f_k^l(\mathbf{r}^l(t)) dt \\
&= \int_0^1 \nabla f_k^l(\mathbf{r}^l(t)) \cdot \frac{d\mathbf{r}^l(t)}{dt} dt \\
&= \sum_{i=1}^{D_l} \int_0^1 g_i^l(t) h_i^l(t) dt,
\end{aligned}
\tag{1}
$$

where

$$g_i^l(t) = \frac{\partial f_k^l(\mathbf{r}^l(t))}{\partial r_i^l(t)} \quad \text{and} \quad h_i^l(t) = \frac{dr_i^l(t)}{dt},$$

with $\cdot$ representing the dot product operator, and $r_i^l(t)$ being the i-th element in the interpolant $\mathbf{r}^l(t)$. The first equality in Eq. 1 consequents from the design of $\mathbf{r}^l$ and the fact that $f^l(\mathbf{x}^l) = f(\mathbf{x})$. The second equality stems from the fundamental theorem of calculus. The third equality arises from the multivariate chain rule, and the last equality results from decomposing the dot product into a summation and then interchanging the order of finite sum and integration.

Equation 1 breaks down the score difference into a sum, where each term is a line integral along the i-th element of curve C^l, and the integrand is a function involving the partial derivative of the prediction $f_k^l(\mathbf{r}^l(t))$ w.r.t. the i-th element in the interpolant $\mathbf{r}^l(t)$. Consequently, each term in the sum resembles the attribution of the prediction $f_k(\mathbf{x})$ to an individual element in $\mathbf{x}^l$ through the integrated partial derivatives along C^l. Equipped with Eq. 1, an explanation map for $\mathbf{x}^l$ can be formed as follows:

$$\mathbf{m}^l = \int_0^1 \mathbf{g}^l(t) \circ \mathbf{h}^l(t) dt, \tag{2}$$

where $\circ$ denotes the element-wise multiplication, $\mathbf{g}^l(t) = \frac{\partial f_k^l(\mathbf{r}^l(t))}{\partial \mathbf{r}^l(t)}$ is the gradient of the prediction w.r.t. the interpolant, and $\mathbf{h}^l(t) = \frac{d\mathbf{r}^l(t)}{dt}$. Note that $\mathbf{m}^l \in \mathbb{R}^{D_l}$ with $m_i^l = \int_0^1 g_i^l(t) h_i^l(t) dt$. Notable, for $l = 0$, Eq. 2 is equivalent to the IG [23] explanation map, where the interpolation takes place in the input space.

Equation 2 integrates the gradients of the interpolated activation maps $\mathbf{r}^l(t)$. Empirically, we found that incorporating the information from $\mathbf{r}^l(t)$ itself (beyond its gradient) yields enhanced explanations, both visually and quantitatively. This observation is consistent with previous works [12, 20, 22]. Furthermore, since for $l > 0$, $\mathbf{m}^l$ does not match the spatial dimensions of the input $\mathbf{x}$, a subsequent transformation ψ^l is employed to ensure a proper match. To this end, we define the DIX explanation map as follows:

$$\mathbf{m}_{\text{DIX}}^l = \psi^l \left(\int_0^1 \phi \left(\mathbf{r}^l(t), \mathbf{g}^l(t) \right) \circ \mathbf{h}^l(t) dt \right), \tag{3}$$

where the exact implementation details of ϕ and ψ are architecture dependent and are outlined in Sec. 4.1.

In this work, we choose C^l to be the linear curve connecting $\mathbf{z}^l$ to $\mathbf{x}^l$, hence

$$\mathbf{r}^l(t) = \mathbf{z}^l + t(\mathbf{x}^l - \mathbf{z}^l) \quad \text{and} \quad \mathbf{h}^l(t) = \mathbf{x}^l - \mathbf{z}^l. \tag{4}$$

In practice, the integration in Eq. 3 is numerically approximated as follows:

$$\mathbf{m}^l_{\text{DIX}} \approx \psi^l \left(\frac{\mathbf{x}^l - \mathbf{z}^l}{N} \circ \sum_{n=1}^{N} \phi \left(\mathbf{r}^l \left(\frac{n}{N} \right), \mathbf{g}^l \left(\frac{n}{N} \right) \right) \right), \tag{5}$$

where we have employed the linear interpolation from Eq. 4. In this work, we set $N = 10$. The complexity of DIX is similar to IG, except for the extra computation induced by ϕ and ψ^l.

Given that different network layers capture varying types of information and resolution, we propose aggregating information from explanation maps produced for different values of l. As such, the final explanation map is constructed as follows:

$$\mathbf{m}^S_{\text{DIX}} = \frac{1}{|S|} \sum_{l \in S} \mathbf{m}^l, \tag{6}$$

where S is a set indicating the layer indexes participating in the aggregation. Our experimentation indicates that the best-performing DIX configurations leverage a combination of explanation maps from the last two or three layers. Thus, in Sec. 6, we report results for $S = \{L-1, L\}$ (**DIX2**) and $S = \{L-2, L-1, L\}$ (**DIX3**). However, for the sake of completeness, we also present results for $S = \{L\}$ (**DIX1**) as part of our ablation study in Sec. 6.4.

4.1 Implementation Details

In this section, we describe concrete implementations of DIX for both CNN and ViT architectures.

CNN Models: In the case of CNNs, the architecture of f consists of residual blocks [49] that produces 3D tensors representing the activation maps $\mathbf{x}^l$. Correspondingly, $\mathbf{z}^l$ is a 3D tensor where each channel is determined by broadcasting the minimum value of the respective activation map within $\mathbf{x}^l$. Furthermore, we set ϕ to the element-wise multiplication.

We motivate this design choice by the fact that $\mathbf{r}^l(t)$ represents the interpolated activation map, highlighting regions where filters are activated and patterns are detected. Its gradient

gauges the attribution degree of the specific class of interest to each element in the activation map. Thus, we expect that regions exhibiting both large gradient and activation (of the same sign) will yield effective explanations. This property is achieved through element-wise multiplication of $\mathbf{r}^l(t)$ by its gradient $\mathbf{g}^l(t)$. Finally, ψ^l is set to the mean reduction on the channel axis followed by a resize operation yielding a 2D explanation map that matches the spatial dimensions of $\mathbf{x}$.

ViT Models: In ViT [5], the architecture of f consists of transformer encoder blocks producing 2D tensors (sequence of token representations). The input $\mathbf{x}$ is transformed to a 2D tensor as well, where the first token is the [CLS] token, and the rest of the tokens are representations of patches in the original image.

In our implementation, we choose to interpolate on the attention matrices, which in turn affect the output produced by the encoder block. Specifically, $\mathbf{r}^l(t)$ is a 3D tensor that accommodates all the attention matrices produced by the l-th encoder block. The reference $\mathbf{z}^l$ is set to the zero tensor (since the values in the attention matrix are in $[0, 1]$). ϕ implements a variant of the Attention Rollout (AR) method [46] that we name Gradient Rollout (GR). GR is similar to AR, with a slight modification. Instead of operating solely on the plain attention matrices, GR initially performs an element-wise multiplication of the attention matrices by their corresponding gradients. Following this, GR proceeds with the original Rollout computation [46], resulting in the first row of the derived matrix (associated with the [CLS] token). This output is further processed by truncating its initial element and reshaping it into a 14×14 matrix. The exact implementation of GR appears in our GitHub repository[1]. Lastly, ψ^l remains consistent across all layers, conducting a resize operation to align with the spatial dimensions of $\mathbf{x}$.

[1] It is worth noting that our experimental findings suggest comparable performance when substituting the matrix product operation with summation within the context of the GR computation

5 Experimental Setup

Our evaluation encompasses three distinct CNN architectures: ResNet101 (**RN**)[2], DenseNet201 (**DN**)[3], and ConvNext-Base (**CN**)[4], and two different architectures of ViT: ViT-Base (**ViT-B**) and ViT-Small (**ViT-S**)[5]. The information regarding preprocessing methodologies and direct access to all the aforementioned models can be found in our GitHub repository. DIX is evaluated and compared to other explanation methods through a series of explanation, segmentation, and sanity tests.

5.1 Explanation Metrics

It is difficult to quantify the quality of explainability methods, and there is no single agreed-upon metric. The explanations metrics in this study aim to assess how well the explanations align with hypothetical changes (counterfactuals) to the input. Essentially, it's about asking "what if" questions regarding the input and determining whether the explanations provided are consistent with those hypothetical scenarios. To comprehensively evaluate our method, we carefully followed several prominent evaluation protocols.

Perturbation Tests We followed the protocol from [13], which is the current state-of-the-art in explaining ViTs, and report the Negative Perturbation AUC (**NEG**) and the Positive Perturbation AUC (**POS**). NEG is a counterfactual test that entails a gradual blackout of the pixels in the original image in increasing order according to the explanation map while searching to see when the model's top predicted class changes. By masking pixels in increasing order, we expect to remove the least relevant pixels first, and the model's top predicted class is expected to remain unchanged for as long as possible. Results are measured in terms of the Area Under the Curve (AUC), and higher values are considered better. Accordingly, the POS test entails masking the pixels in decreasing order with the expectation that the model's top predicted class will change quickly, hence in POS, lower values are better. In addition, we follow [50] and report the Insertion AUC (**INS**) and Deletion AUC (**DEL**) perturbation tests. INS and DEL entail a gradual blackout in increasing or decreasing order, similar to NEG and POS, respectively. However instead of tracking the point at which the top predicted class changes, in **INS** and DEL the AUC is computed with respect to the predicted probability of the top class. By masking pixels according to increasing/decreasing order of importance, we expect that the predicted probability of the top class will decrease slowly/quickly, respectively. Hence, for INS higher values are better and for DEL lower values are better.

ADP and PIC Tests We follow [20] and report the Average Drop Percentage (**ADP**) and the Percentage Increase in Confidence (**PIC**) tests. Both tests relate to the change in the probability of the predicted class after applying the mask to the original image. A good explanation map is expected to highlight the most significant regions for decision-making. Hence, applying such a mask can be seen as a removal of the "background". The ADP test measures the average percentage of model confidence drop after applying the mask. A good mask is expected to maintain the most relevant areas and minimize confidence drop, hence for ADP lower values are considered better. However, we note that ADP is a problematic metric since a naive all-ones mask yields an optimal ADP value of 0. Nevertheless, we included it for the sake of compatibility with previous works [20]. In some instances, the model's confidence increases after applying a good explanation mask that removes a confusing background. Hence, PIC is a binary test that measures the percentage of instances in which the model's confidence increased after applying the mask on the original input. For PIC higher values are considered better.

AIC and SIC Tests We follow [51] and report the Accuracy Information Curve (**AIC**) and the Softmax Information Curve (**SIC**) tests. In these tests, we start with a completely blurred image and gradually sharpen the image areas that are deemed important by a given explanation method. Gradually sharpening the image areas increases the information content of the image. We then compare the explanation methods by measuring the approximate image entropy (e.g., compressed image size) and the model's performance (e.g., model accuracy). The AIC metric measures the accuracy of a model as a function of the amount of information provided to the explanation method. AIC is defined as the AUC of the accuracy vs. information plot. The SIC metric measures the information content of the output of a softmax classifier as a function of the amount of information provided to the explanation method. SIC is defined as the AUC of the entropy vs. information plot. The entropy of the softmax output is a measure of the uncertainty or randomness of the classifier's predictions. For both AIC and SIC, the information provided to the method is quantified by the fraction of input features that are considered during the explanation process.

5.2 Segmentation Metrics

While possessing a superior segmentation capability does not necessarily imply a superior explanatory aptitude, we undertake this evaluation task for the sake of completeness in our comparison with previous works assessing this aspect [13, 26, 21, 52]. Segmentation accuracy is assessed according to the following metrics: Pixel Accuracy (**PA**), mean-

intersection-over-union (**mIoU**), mean-average-precision (**mAP**), and the mean-F1 score (**mF1**) [13].

5.3 Datasets

Explanation maps are produced for the ImageNet [53] ILSVRC 2012 (**IN**) validation set, consisting of 50K images from 1000 classes. We follow the same setup from [13], where for each image, an explanation map is produced w.r.t. the class predicted by the model. Segmentation tests are conducted on three datasets: (1) ImageNet-Segmentation [54] (**IN-Seg**): This is a subset of ImageNet validation set consisting of 4,276 images from 445 classes for which annotated segmentations are available. (2) Microsoft Common Objects in COntext 2017 [55] (**COCO**): This is a validation set that contains 5,000 annotated segmentation images from 80 different classes. Some images consist of multi-label annotations (multiple annotated objects). In our evaluation, all annotated objects in the image are considered as the ground-truth. (3) PASCAL Visual Object Classes 2012 [56] (**VOC**): This is a validation set that contains annotated segmentations for 1,449 images from 20 classes.

5.4 Evaluated Methods

Our evaluation encompasses a comprehensive assessment of various explanation methods, including gradient-based approaches, path-integration techniques, as well as gradient-free methods.

For CNN models, the following explanation techniques are considered: Integrated Gradients (**IG**) [23], Guided IG (**GIG**) [44], Blur IG (**BIG**) [43], Ablation-CAM (**AC**) [39], Layer-CAM (**LC**) [21], LIFT-CAM (**LIFT**) [40], Grad-CAM (**GC**) [12], Grad-CAM++ (**GC++**) [20], X-Grad-CAM (**XGC**) [37], and FullGrad (**FG**) [35].

For ViT models, we consider two state-of-the-art methods: Transformer Attribution (**T-Attr**) [13] and Generic Attention Explainability (**GAE**) [26]. Both methods were shown to outperform other strong baselines such as partial LRP [48], and GC [26] for transformers. A detailed description of all explanation methods is provided in our GitHub repository. Lastly, our universal DIX method is evaluated on both CNNs and ViTs, where we consider two versions: DIX2 and DIX3 following the description in Sec. 4. In what follows, we briefly describe the evaluated methods.

- Grad-CAM (**GC**) [12] integrates the activation maps from the last convolutional layer in the CNN by employing global average pooling on the gradients and utilizing them as weights for the feature map channels.

- Grad-CAM++ (**GC++**) [20] is an advanced variant of Grad-CAM that utilizes a weighted average of the pixel-wise gradients to generate the activation map weights.

- XGrad-CAM (**XGC**) [37] calculates activation coefficients using two axioms. Although the authors derived coefficients that satisfy these axioms as closely as possible, their derivation is only demonstrated for ReLU-CNNs.

- Integrated Gradients (**IG**) [23] integrates over the interpolated image gradients.

- Blur IG (**BIG**) [43] is concerned with the introduction of information using a baseline and opts to use a path that progressively removes Gaussian blur from the attributed image.

- Guided IG (**GIG**) [44] improves upon Integrated Gradients by introducing the idea of an adaptive path method. By calculating integration along a different path than Integrated Gradients, high gradient areas are avoided which often leads to an overall reduction in irrelevant attributions.

- LIFT-CAM (**LIFT**) [40] employs the DeepLIFT [41] technique to estimate the activation maps SHAP values [42] and then combine them with the activation maps to produce the explanation map.

- The FullGrad (**FG**) method [35] provides a complete modeling approach of the gradient by also taking the gradient with respect to the bias term, and not just with respect to the input.

- LayerCAM (**LC**) [21] utilizes both gradients and activations, but instead of using the Grad-CAM approach and applying pooling on the gradients, it treats the gradients as weights for the activations by assigning each location in the activations with an appropriate gradient location. The explanation map is computed with a location-wise product of the positive gradients (after ReLU) with the activations, and the map is then summed w.r.t. the activation channel, with a ReLU applied to the result.

- Ablation-CAM (**AC**) [39] is an approach that only uses the channels of the activations. It takes each activation channel, masks it from the final map by zeroing out all locations of this channel in the explanation map produced by all the channels, computes the score on the masked explanation map (the map without the specific

channel), and this score is used to assign an importance weight for every channel. At last, a weighted sum of the channels produces the final explanation map.

- The Transformer attribution (**T-ATTR**) [13] method computes the importance of each input token by analyzing the attention weights assigned to it during self-attention. Specifically, it computes the relevance score of each token as the sum of its attention weights across all layers of the Transformer. The intuition behind this approach is that tokens that receive more attention across different layers are likely more important for the final prediction. To obtain a more interpretable and localized visualization of the importance scores, the authors also propose a variant of the method called Layer-wise Relevance Propagation (LRP), which recursively distributes the relevance scores back to the input tokens based on their contribution to the intermediate representations.

- Generic Attention Explainability (**GAE**) [26] is a generalization of T-Attr for explaining Bi-Modal transformers.

5.5 Sanity Tests for Explanation Methods

To comprehensively assess the robustness and credibility of DIX, we conducted the *parameter randomization* and *data randomization* sanity tests as outlined in [27]. For these evaluations, we employed DIX3, along with the VGG-19[57] model and the IN dataset.

5.5.1 Parameter Randomization Test

The parameter randomization test compares the explanation maps produced by the explanation method based on two setups of the same model architecture: (1) Trained - the model is trained on the dataset (e.g., a pretrained VGG-19 model that was trained on ImageNet, and (2) Random - the same model architecture, with random weights (e.g., a randomly initialized VGG-19 model). For a method that relies on the actual model to be explained, we anticipate significant differences in the explanation maps produced for the trained model and those produced for the random model. Conversely, if the explanation maps are similar, we conclude that the explanation method is insensitive to the model's parameters, and thus may not be useful for explaining and debugging the model.

Given a trained model, we consider two types of parameter randomization tests: The first test randomly re-initializes all weights of the model in a cascading fashion (layer after layer). The second test independently randomizes one layer at a time, while keeping all

other layers fixed. In both cases, we compare the resulting explanations obtained by using the model with random weights to those derived from the original weights of the model.

Cascading Randomization The cascading randomization method involves the randomization of a model's weights, starting from the top layer and successively moving down to the bottom layer. This process leads to the destruction of the learned weights from the top to the bottom layers.

Independent Randomization We further consider another version of the model's parameters randomization test, in which a layer-by-layer randomization is employed, one layer at a time. In this test, we aim to isolate the influence of the randomization of each layer, hence randomization is applied to one layer's weights at a time, while all other layers' weights are kept identical to their values in the original model. This randomization methodology enables comprehensive evaluation of the sensitivity of the explanation maps w.r.t. each of the model's layers.

5.5.2 Data Randomization Test

The data randomization sanity test is a method used to assess whether an explanation method is sensitive to the labeling of the data used for training the model. This is done by comparing the explanation maps produced by the explanation method for two models with identical architecture that were trained on two different datasets: one with the original labels and another with randomly permuted labels. If the explanation method is sensitive to the labeling of the dataset, we would expect the produced explanation maps to differ significantly between the two cases. However, if the method is insensitive to the permuted labels, it indicates that it does not depend on the relationship between instances and labels that exists in the original data. To conduct the data randomization test, we permute the training labels in the dataset and train the model to achieve a training set accuracy greater than 95%. Note that the resulting model's test accuracy is never better than randomly guessing a label. We then compute explanations on the same test inputs for both the model trained on true labels and the model trained on randomly permuted labels.

6 Results

6.1 Explanation Tests

Tables 1 and 2 provide a comprehensive explanation tests for CNN and ViT models, respectively. We report results for all combinations of datasets, models, methods, and metrics. Our analysis demonstrates that DIX consistently surpasses all baseline methods across a spectrum of metrics and architectural configurations. On CNN-based DIX variations (Tab. 1), DIX3 outclasses DIX2 in terms of NEG, INS, SIC, and AIC metrics for both RN and DN backbones, while demonstrating dominance across all metrics for the CN backbone. Regarding the ViT-based DIX variants (Tab. 2), DIX3 outperforms DIX2 across all metrics (with the exception of PIC on ViT-B, and PIC and ADP on ViT-S).These trends showcase the advantage of aggregating information from more layers. In the context of CNNs, the second-best performing methods are GC and GC++, which leverage both activation and gradients to outperform other approaches across most evaluation metrics. Additionally, we note that path integration techniques (IG, BIG, and GIG) demonstrate competitive results in terms of POS and DEL metrics, while displaying comparatively weaker performance in other aspects. This disparity may be attributed to the grainy output maps generated by path integration techniques, as evidenced in Fig.3 for IG explanation maps on CNNs. These methods ignore the activations and integrate on the image domain only, hence missing some of the key features. This is particularly evident in the significant contrast between their strong performance on POS and the corresponding weaker performance on NEG. As path integration methods produce sparse maps that can negatively affect performance in certain metrics, , we extend our analysis to encompass the SIC and AIC metrics as well [51]. These metrics were originally employed to assess GIG[44] and BIG[43]. Yet, the incorporation of SIC and AIC did not alter the trend of the results. This suggests that DIX is highly effective for generating high-quality explanation maps. Finally, we present an ablation study in Section 6.4, aimed at comparing diverse versions and alternatives of DIX. This analysis serves to emphasize the effectiveness of the integration process and the strategic utilization of information from multiple layers within the DIX methodology.

6.2 Segmentation Tests

Tables 3 and 4 present segmentation tests results on CNN and ViT models, respectively. The results are reported for all combinations of datasets, models, explanation methods, and segmentation metrics. In these experiments, only the 5 best performing CNN explanation

Table 1: Explanation tests results on the IN dataset (CNN models): For POS, DEL and ADP, lower is better. For NEG, INS, PIC, SIC and AIC, higher is better. See Sec. 6.1 for details.

		GC	GC++	LIFT	AC	IG	GIG	BIG	FG	LC	XGC	DIX2	DIX3
RN	NEG	56.41	55.20	55.39	54.98	45.66	43.97	42.25	54.81	53.52	53.46	56.28	**57.13**
	POS	17.82	18.01	17.53	19.38	17.24	17.68	17.44	18.06	17.92	21.02	**15.69**	17.11
	INS	48.14	47.56	45.39	47.05	39.87	37.92	36.04	42.68	46.11	43.26	48.09	**48.91**
	DEL	13.97	14.17	15.32	14.23	13.49	14.18	13.95	14.64	14.31	14.98	**12.84**	13.36
	ADP	17.87	16.91	18.03	16.18	37.52	35.28	40.85	21.06	24.34	17.02	**15.68**	16.02
	PIC	36.69	36.53	35.95	35.52	19.94	18.72	24.53	31.59	35.43	36.18	**40.21**	37.29
	SIC	76.91	76.44	76.73	73.36	54.67	55.04	56.98	75.35	73.93	72.64	77.61	**78.12**
	AIC	74.36	71.97	72.76	70.35	51.92	53.38	53.36	71.49	65.77	69.85	76.09	**76.34**
CN	NEG	52.86	53.82	53.98	53.68	45.24	41.43	40.72	52.06	54.12	52.13	54.40	**55.23**
	POS	17.52	17.85	18.23	18.19	17.42	18.03	18.14	18.26	17.58	20.83	16.96	**16.51**
	INS	45.65	45.19	43.86	49.18	37.22	32.99	31.02	42.01	44.14	42.07	49.53	**49.86**
	DEL	13.43	14.17	15.18	14.73	12.36	13.08	13.29	14.21	13.64	14.78	11.95	**11.74**
	ADP	22.46	22.35	29.13	24.38	36.98	35.79	41.73	30.75	37.62	25.68	22.24	**22.19**
	PIC	23.16	24.42	22.34	24.59	17.65	13.12	20.69	22.13	22.17	23.26	28.31	**28.47**
	SIC	65.93	67.94	54.75	63.95	53.36	58.35	57.27	62.84	69.11	59.12	69.83	**70.18**
	AIC	75.64	75.52	57.06	71.53	51.68	55.82	53.82	67.15	75.41	62.38	76.44	**77.29**
DN	NEG	57.40	57.16	58.01	56.63	40.74	37.31	36.67	56.79	56.96	55.74	57.31	**58.25**
	POS	17.75	17.81	18.87	18.67	17.31	17.46	17.38	17.84	17.62	18.67	**16.59**	17.14
	INS	51.09	50.89	50.63	50.41	37.58	33.31	31.32	50.44	50.60	49.62	50.97	**51.58**
	DEL	13.61	13.63	13.29	15.31	13.26	13.27	13.54	14.34	13.85	14.75	**12.73**	12.98
	ADP	17.46	17.01	19.45	17.13	35.61	34.51	40.04	20.21	24.23	19.59	**16.29**	16.58
	PIC	34.68	35.21	34.13	31.22	22.35	16.62	26.18	31.05	33.81	30.39	**38.91**	37.78
	SIC	75.62	74.75	74.72	73.94	54.59	58.55	57.66	72.93	74.34	73.94	77.24	**77.32**
	AIC	74.22	71.82	72.65	70.21	54.74	54.56	56.08	70.63	71.82	70.12	75.98	**76.39**

methods from Tab. 1 are considered. Once again, it becomes evident that DIX consistently delivers the most favorable segmentation outcomes for both CNN and ViT models. This outcome can be rationalized by the localized and precise maps that DIX generates.

6.3 Qualitative Evaluation

Figure 1 presents a qualitative comparison of the explanation maps obtained by the top-performing CNN explanation methods on a large set of examples that are randomly drawn from multiple classes from the IN dataset. Arguably, DIX (DIX3) produces the most accurate explanation maps in terms of class discrimination and localization. These results correlate well with the trends from Tabs. 1 and 3. We observe that in the case of class 'accordion, piano accordion, and squeeze box', DIX focuses mostly on the correct item, while the gradient-free methods focus mostly on other parts of the image, exposing their class-agnostic behavior. Moreover, a similar trend is observed with the 'sturgeon' class, in which DIX is the only one to focus on the relevant class. Figure 2 presents a qualitative comparison of the explanation maps obtained by explanation methods for ViT. Once again,

Table 2: Explanation tests results on the IN dataset (ViT models): For POS, DEL and ADP, lower is better. For NEG, INS, PIC, SIC and AIC, higher is better. See Sec. 6.1 for details.

		T-Attr	GAE	DIX2	DIX3
ViT-B	NEG	54.16	54.61	56.43	**56.94**
	POS	17.03	17.32	15.10	**14.85**
	INS	48.58	48.96	49.51	**50.59**
	DEL	14.20	14.37	12.62	**12.16**
	ADP	54.02	37.84	35.93	**35.58**
	PIC	13.37	23.65	**28.21**	27.41
	SIC	68.59	68.35	68.94	**69.11**
	AIC	61.34	57.92	62.42	**65.03**
ViT-S	NEG	53.29	52.81	55.98	**56.13**
	POS	14.16	14.75	13.09	**12.32**
	INS	45.72	45.21	46.62	**47.36**
	DEL	11.28	11.92	11.18	**10.56**
	ADP	51.94	36.98	**36.31**	36.57
	PIC	13.67	8.68	**18.39**	18.25
	SIC	69.46	70.19	70.92	**71.55**
	AIC	63.86	64.49	65.17	**65.58**

we see that DIX produces the most accurate and focused explanation maps.

6.4 Ablation Study

In this work we present and evaluate DIX2 and DIX3. In this section, we justify these choices via an ablation study. To this end, we set $n = 10$, and consider three alternatives: (1) **DIX1** - we use the last layer as the only layer to interpolate i.e., $S = \{L\}$. (2) **DIX2-MUL** - $\mathbf{m}^{L}_{\text{DIX}}$ and $\mathbf{m}^{L-1}_{\text{DIX}}$ are being element-wise multiplied to produce the final explanation map. (3) **DIX3-GRADS** - we use the plain gradients without explicitly incorporating the information from the activation or attention maps.

Table 5 reports the results for the RN and ViT-B models on the IN dataset. For the sake of completeness, we further include the results for IG, DIX2, and DIX3 (taken from Tabs. 1 and 2). First, we can see the superior performance of DIX2 and DIX3 across all metrics and models. We further observe that both DIX1 and DIX2-MUL fall short in comparison to DIX2. This observation underscores the inherent necessity of incorporating information from additional layers and shows the advantages of aggregation via summation. When aggregating the explanation maps of different layers, the objective is to effectively incorporate data from each map to capture a richer spectrum of insights and class-specific signals. Notably, the multiplication operator exhibits a behavior akin to intersection, where both high pixel

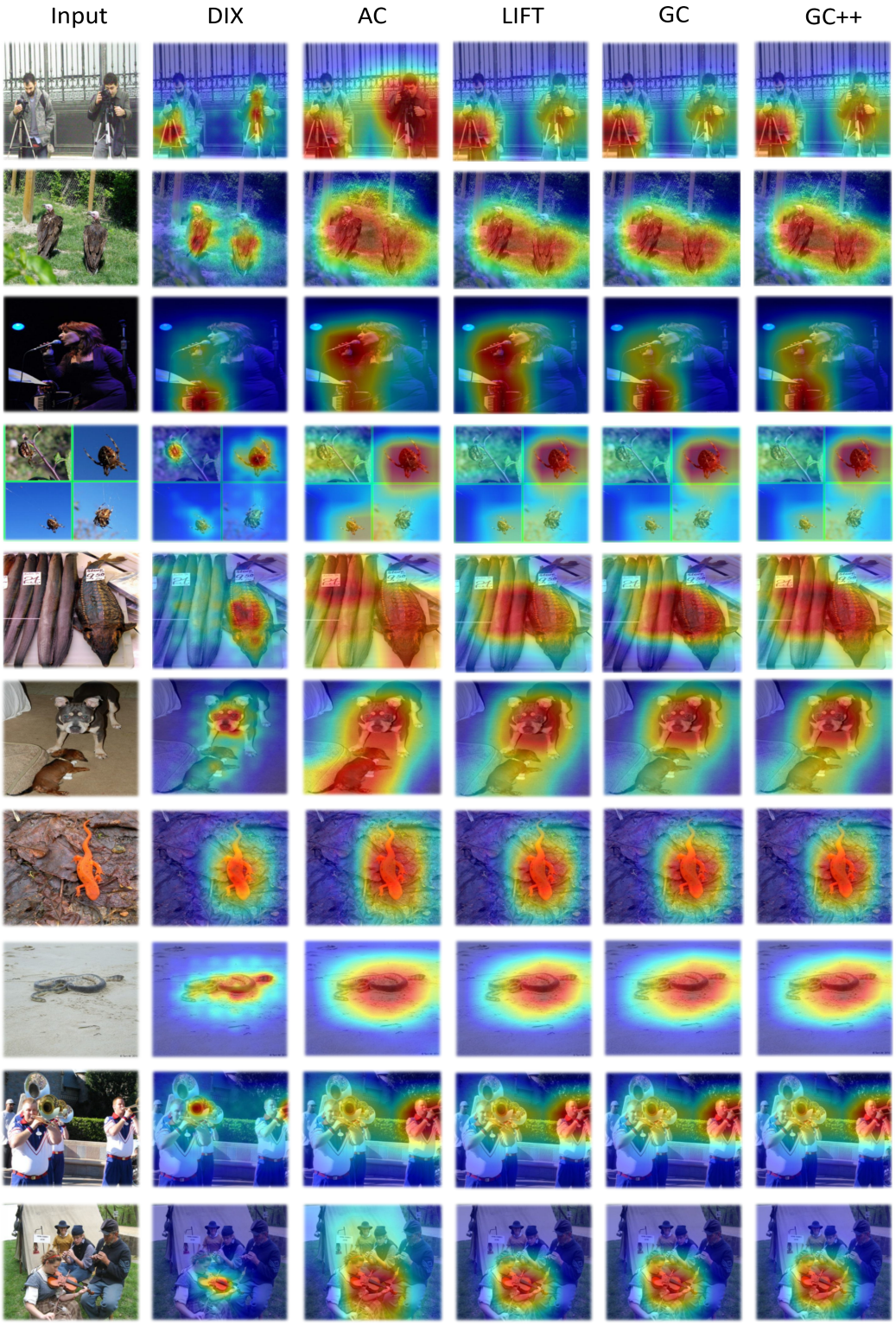

Figure 1: CNN Qualitative Results: Explanation maps produced using RN w.r.t. the classes (top to bottom): 'tripod', 'vulture', 'accordion, squeeze box', 'garden spider, Aranea diademata', 'sturgeon', 'American Staffordshire terrier, Staffordshire terrier, American pit bull terrier, pit bull terrier', 'eft', 'sea snake', 'trombone' and 'violin, fiddle'.

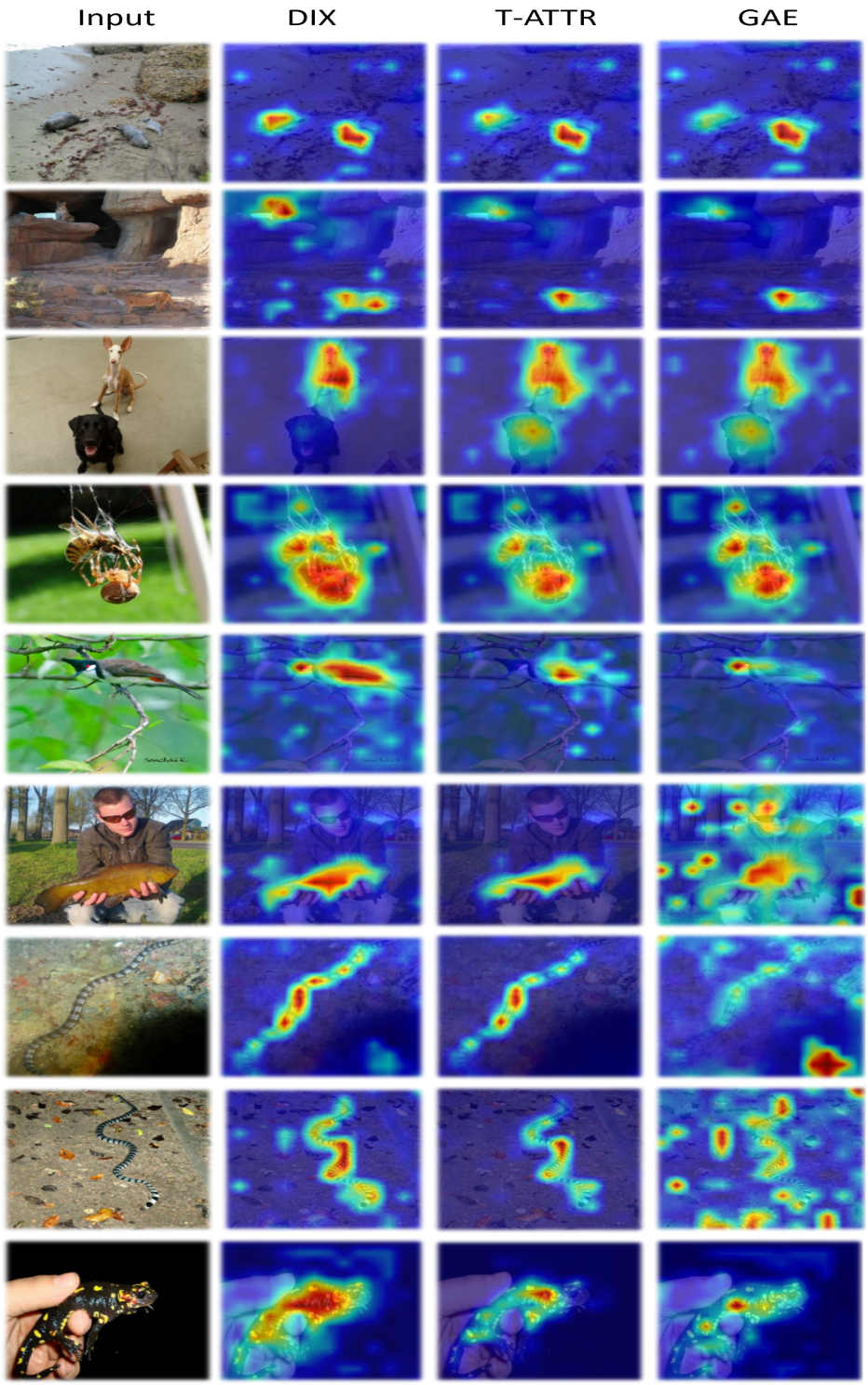

Figure 2: ViT Qualitative Results: Explanation maps produced using ViT-B w.r.t. the classes (top to bottom): 'sea lion', 'cougar, puma, catamount, mountain lion, painter, panther, Felis concolor', 'Ibizan hound, Ibizan Podenco', 'garden spider, Aranea diademata', 'bulbul', 'tench, Tinca tinca', 'sea snake', 'night snake, Hypsiglena torquata' and 'European fire salamander, Salamandra salamandra'.

Table 3: Segmentation tests on three datasets (CNN models). For all metrics, higher is better. See Sec. 6.2 for details.

			GC	GC++	LIFT	AC	DIX2	DIX3
IN-SEG	CN	PA	77.01	77.54	63.77	77.04	78.32	**78.93**
		mAP	81.01	85.63	69.40	86.93	87.13	**87.34**
		mIoU	56.58	58.35	53.81	58.42	58.64	**58.79**
		mF1	36.88	38.26	35.91	41.29	42.51	**42.95**
	RN	PA	71.93	71.96	71.68	70.36	72.43	**73.17**
		mAP	84.21	84.23	83.79	81.14	84.58	**85.37**
		mIoU	53.06	53.29	52.17	52.91	53.93	**54.16**
		mF1	42.51	42.68	41.95	42.08	42.75	**43.18**
	DN	PA	73.00	73.21	72.87	72.44	73.58	**73.90**
		mAP	85.04	85.53	84.82	84.62	85.57	**85.98**
		mIoU	54.18	54.57	54.11	54.89	55.42	**56.03**
		mF1	41.74	42.58	41.61	43.51	43.71	**43.79**
COCO	CN	PA	68.75	66.49	60.37	64.10	68.87	**69.38**
		mAP	75.02	75.21	67.98	76.09	76.94	**77.43**
		mIoU	43.46	44.01	37.08	44.27	44.89	**45.06**
		mF1	28.96	29.85	26.92	30.81	31.28	**31.99**
	RN	PA	64.17	64.39	64.02	63.90	64.75	**64.94**
		mAP	74.19	74.27	73.78	72.80	74.38	**74.91**
		mIoU	42.37	43.25	42.59	42.88	43.54	**43.87**
		mF1	31.64	32.82	31.77	32.41	33.39	**33.71**
	DN	PA	63.50	64.06	63.25	64.51	64.98	**65.37**
		mAP	72.61	73.07	72.15	73.85	74.02	**74.67**
		mIoU	43.02	43.75	42.85	44.16	44.75	**44.82**
		mF1	31.04	32.31	30.83	33.93	34.14	**34.59**
VOC	CN	PA	72.54	72.09	63.32	69.83	72.68	**72.81**
		mAP	77.27	79.47	68.83	80.45	81.35	**81.79**
		mIoU	50.28	50.63	48.86	49.76	51.12	**51.29**
		mF1	35.24	35.67	33.26	34.51	35.92	**36.57**
	RN	PA	68.74	69.01	68.61	68.00	69.38	**69.74**
		mAP	79.68	79.96	79.41	78.02	81.02	**81.49**
		mIoU	49.44	49.91	49.15	49.32	50.43	**51.58**
		mF1	33.08	33.56	32.69	32.74	34.28	**34.68**
	DN	PA	68.43	68.78	68.24	68.36	68.89	**68.95**
		mAP	78.68	79.06	78.52	78.62	79.43	**79.66**
		mIoU	49.29	49.68	49.03	49.11	49.91	**50.24**
		mF1	32.92	33.83	32.28	32.56	34.11	**34.26**

values are required for proper appearance in the final map. This characteristic, as depicted in Figure 3, contrasts with the intended outcome. Furthermore, the superiority of DIX3 over DIX3-GRADS underscores the benefit from exploiting intermediate representation information alongside its corresponding gradients, which contributes to the generation of

Table 4: Segmentation tests on three datasets (ViT models). For all metrics, higher is better. See Sec. 6.2 for details.

			T-Attr	GAE	DIX2	DIX3
IN-Seg	ViT-B	PA	79.70	76.30	79.91	**81.02**
		mAP	86.03	85.28	87.12	**87.45**
		mIoU	61.95	58.34	62.53	**63.47**
		mF1	40.17	41.85	44.94	**45.66**
	ViT-S	PA	80.86	76.66	81.54	**81.83**
		mAP	86.13	84.23	86.48	**86.96**
		mIoU	63.61	57.70	64.13	**64.67**
		mF1	43.60	40.72	46.34	**46.82**
COCO	ViT-B	PA	68.89	67.10	68.95	**69.42**
		mAP	78.57	78.72	80.63	**81.22**
		mIoU	46.62	46.51	47.75	**47.79**
		mF1	26.28	31.70	33.87	**34.12**
	ViT-S	PA	69.90	67.95	70.41	**70.64**
		mAP	79.28	78.65	80.55	**80.89**
		mIoU	48.62	46.52	50.81	**51.22**
		mF1	30.88	30.96	35.61	**35.74**
VOC	ViT-B	PA	73.70	71.32	75.33	**75.84**
		mAP	81.08	80.88	81.75	**81.89**
		mIoU	53.09	51.82	53.62	**53.71**
		mF1	31.50	35.72	36.38	**36.59**
	ViT-S	PA	74.96	71.85	76.35	**76.56**
		mAP	81.76	80.60	82.74	**82.91**
		mIoU	55.37	51.55	55.83	**55.98**
		mF1	36.03	34.95	39.27	**39.41**

localized, accurate and class discriminative explanation maps. The results presented in Table 5 highlight a distinct advantage for IG and DIX2-MUL with respect to the POS and DEL metrics when compared to DIX3-GRADS and DIX1, both of which generate less concentrated explanation maps. This is due to the fact that the deletion of the most relevant pixels results in fewer pixels being removed, and the mask is more focused on a subset of pixels. DIX1, for instance, produces less focused explanation maps that may highlight irrelevant areas. Such coarse highlighting leads to a slower decrease in the prediction score during the deletion process. On the contrary, DIX1 and DIX3-GRADS exhibit superior performance in relation to the NEG and INS metrics. This divergence in performance can be attributed to the expansive nature of their explanation map, resulting in numerous pixels that require removal. In the context of the NEG metric, this characteristic contributes to a slow decrease in the prediction score during the deletion process and, subsequently, a larger area under the curve (AUC).

Table 5: Ablation study results for various DIX configurations on the IN dataset. See Sec. 6.4 for details.

		DIX1	IG	DIX2	DIX2-MUL	DIX3-GRADS	DIX3
RN	NEG	55.47	45.66	56.28	55.24	56.05	**57.13**
	POS	17.47	17.24	**15.69**	17.28	18.13	17.11
	INS	47.53	39.87	48.09	47.13	47.88	**48.91**
	DEL	13.72	13.49	**12.84**	13.59	14.52	13.36
	ADP	17.21	37.52	**15.68**	21.38	17.43	16.02
	PIC	36.54	19.94	**40.21**	28.46	37.10	37.29
	SIC	76.85	54.67	77.61	75.17	76.13	**78.12**
	AIC	75.48	51.92	76.09	74.21	74.88	**76.34**
ViT-B	NEG	55.98	40.94	56.43	55.62	55.78	**56.94**
	POS	15.49	22.43	15.10	15.37	15.81	**14.85**
	INS	49.38	35.07	49.51	49.27	49.33	**50.59**
	DEL	13.06	17.90	12.62	12.85	13.12	**12.16**
	ADP	36.96	41.35	35.93	38.62	36.08	35.58
	PIC	26.94	16.89	**28.21**	26.39	27.13	**27.41**
	SIC	67.79	58.91	68.94	68.43	68.32	**69.11**
	AIC	61.56	54.93	62.42	62.18	61.94	**65.03**

6.5 Sanity Tests

Unless stated otherwise, the experiments utilize the ImageNet ILSVRC 2012 validation set [53] with the VGG-19 [57] model and DIX3. In what follows, we show that DIX passes all sanity tests success- fully, thereby furnishing additional substantiation for the authenticity of DIX as a robust machinery for generating accurate expla- nation maps.

6.5.1 Parameter Randomization Test

Cascading Randomization Figure 4 presents the Spearman correlation (averaged on 50K examples) between the original explanation map obtained by DIX and the original (pretrained) VGG-19 model and the explanation map obtained by DIX and each of the cascade randomization versions of the original model. The markers on the x-axis are between '0' and '16', where $x = k$ means that the weights of the last k layers of the model are randomized. At $x = 0$ there is no randomization, hence the correlation with the original model is perfect. Starting from $x = 1$ (marked by the horizontal dashed line) and up to $x = 16$, the graph depicts a progressive cascade randomization of the original model. We observe that as more layers' weights are randomized, the correlation with the explanation map of the original model significantly deteriorates. This behavior showcases the sensitivity of DIX to the model's parameters - an expected and desired property for any explanation method [27].

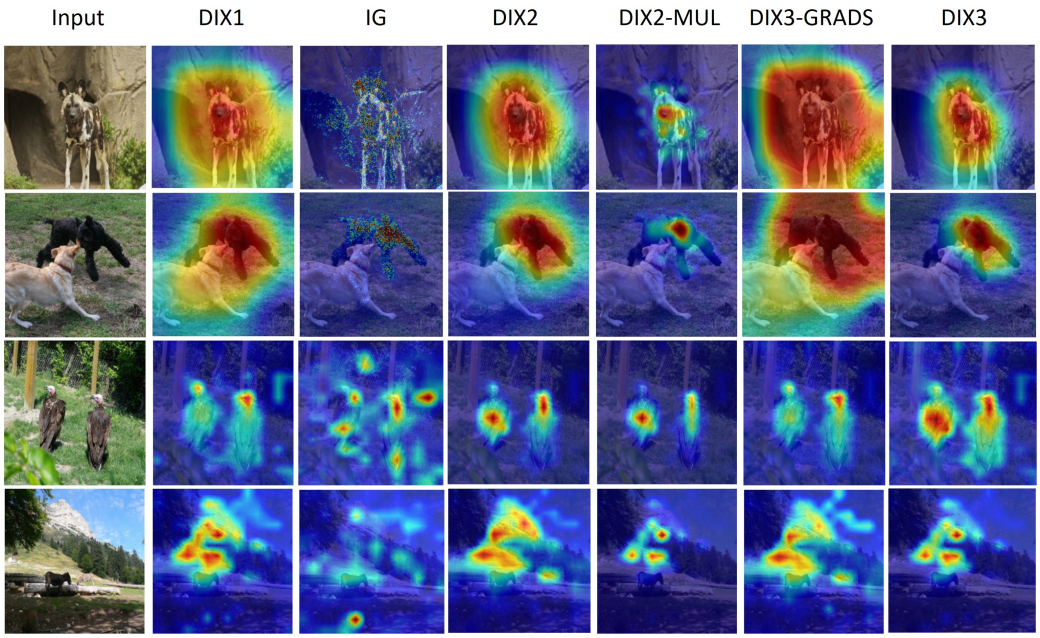

Figure 3: Ablation study results. Explanation maps produced using RN (rows 1,2) and ViT-B (rows 3,4) w.r.t. the classes (top to bottom): 'African hunting dog, hyena dog, Cape hunting dog, Lycaon pictus', 'Kerry blue terrier', 'vulture', 'alp'.

Figure 5 displays a representative example of explanation maps (bottom) and their overlay to the original image (top), illustrating the cascading randomization process. The first column presents explanation maps produced by DIX and the original model, while the rest of the columns present explanation maps produced by DIX and cascading randomized models, where the number i above each column indicates that the explanation map is produced by a model in which the weights of the last i layers were randomized. It is evident that the quality of produced explanation maps significantly degrades as more and more layers are set with random weights.

Independent Randomization Figure 6 presents results for the independent randomization tests. At $x = 0$ no randomization was applied and the correlation to the original model is perfect. For $x = i$ $(i > 0)$ the graph indicates the correlation of the original model with a model in which only the weights of the i-th penultimate layer were randomized while the weights of all other layers were kept untouched. We observe that the correlation values are low across all layers which indicates DIX's sensitivity to weight randomization in each layer separately. This property is a desired property for an explanation method, as it indicates the method's sensitivity to each of the model's layers, independently. Finally, Fig. 7

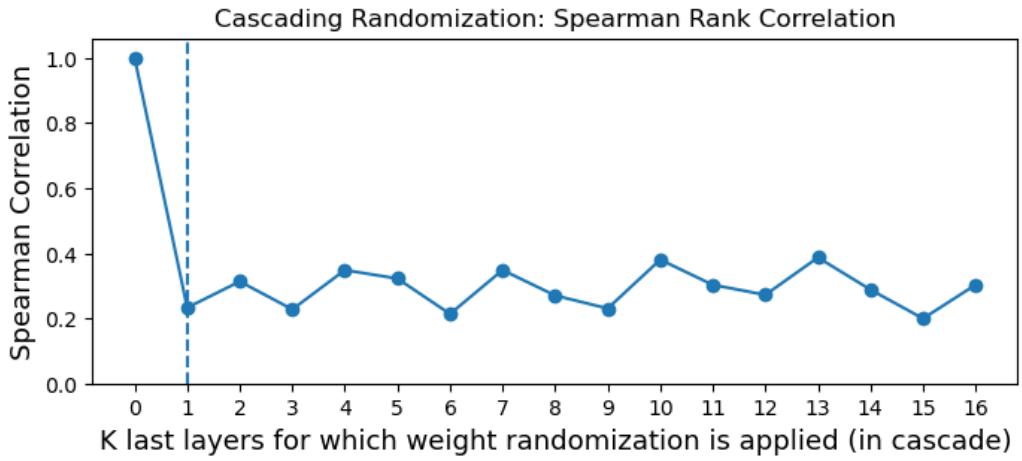

Figure 4: **Cascading Randomization**: The VGG-19 model is subjected to successive weights randomization, beginning from the last model's layers on the ImageNet dataset. The presented graph depicts the Spearman rank correlation (averaged on 50K examples) between the explanation produced by DIX using the original and randomized model's weights. The x-axis corresponds to the number of layers being randomized, starting from the output layer. The dashed line indicates the point where the successive randomization of the network commences, which is at the top layer. The first dot (x=0) corresponds to no randomization (the original model is used), hence the correlation between the explanation maps is perfect. See Sec. 6.5.1 for further details.

presents a qualitative example in the same fashion as Fig. 5, this time for the independent randomization test. We observe that the quality of all explanation maps produced by a randomized version of the model differs significantly from the original explanation map. We conclude that DIX successfully passes both types of parameter randomization tests.

6.5.2 Data Randomization Test

Figure 8 presents a box plot computed for the Spearman correlation values obtained for paired explanation maps (50K examples): one produced using the original model that is trained with the ground truth, and another produced by the model trained with the permuted labels. We can see that the correlation values are very low indicating DIX's sensitivity to the labeling of the dataset. Hence, we conclude that DIX successfully passes the data randomization test.

Finally, Figure 9 presents additional qualitative examples for both tests, this time with different models. The first row shows two explanation maps produced by DIX w.r.t. the "tabby cat" class. We see that when DIX utilizes an ImageNet pretrained ResNet50 model,

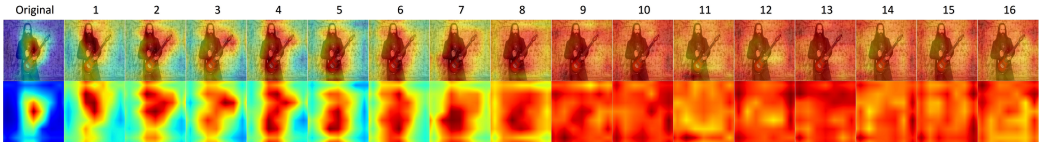

Figure 5: **Cascading Randomization on VGG-19 (ImageNet)**: The figure presents the original explanations (first column) for 'electric guitar'. Progression from left to right depicts the gradual randomization of network weights up to the layer number depicted at the top of the column (starting from the last layer). See Sec. 6.5.1 for further details.

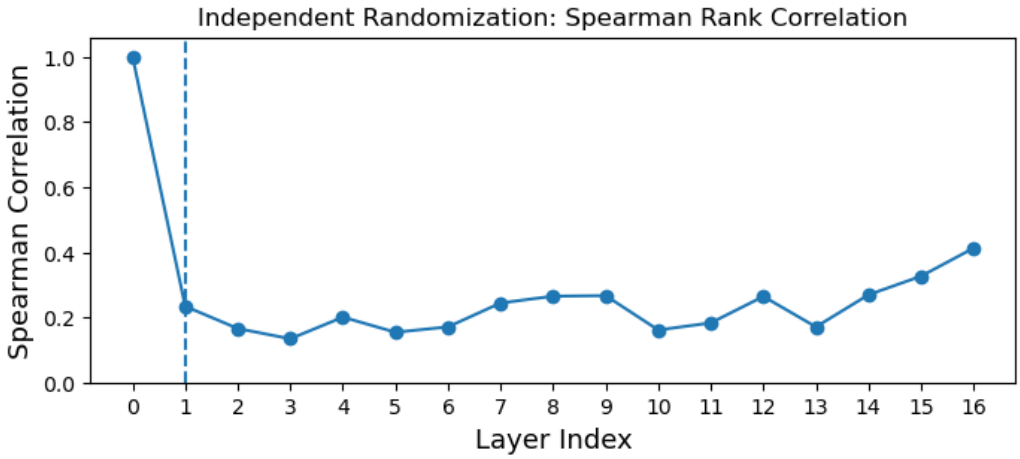

Figure 6: **Independent Randomization**: The randomization process is carried out independently for each layer of the model, while the remaining weights are retained at their pretrained values. The y-axis of the presented graph represents the rank correlation between the original and randomized explanations, with each point on the x-axis corresponding to a specific layer of the model. The dashed line marks the point where the randomization of the network layers commences, which is at the top layer.

it produces a focused explanation map (around the cat), but when applying DIX to the same model with random weights, it fails to detect the cat in the image. The second row shows that DIX produces an adequate explanation map when the model (LeNet-5 [58]) is trained with the MNIST ground truth labels but fails when the model is trained with random labels.

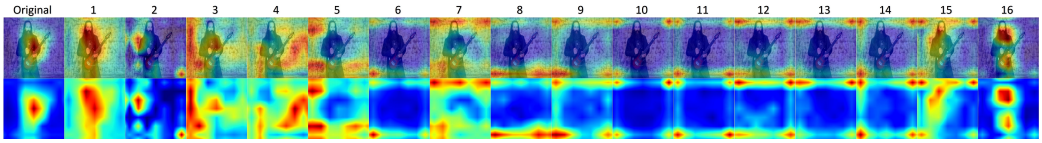

Figure 7: **Independent Randomization on VGG-19 (ImageNet)**: Similar to Fig. 5, however, this time, each specific layer is randomized independently, while the rest of the weights are kept at their pretrained values.

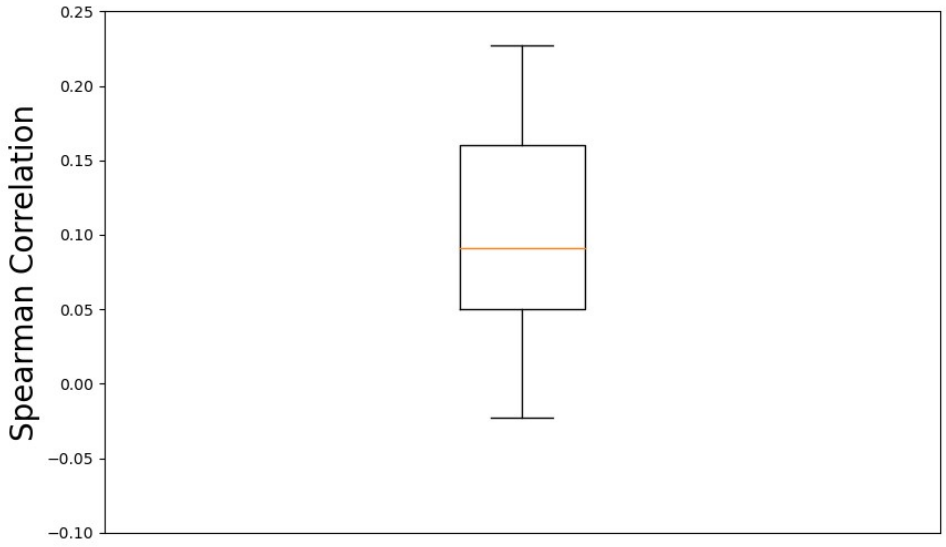

Figure 8: **Data Randomization Test**: Spearman rank correlation box plot for DIX with the VGG-19 model.

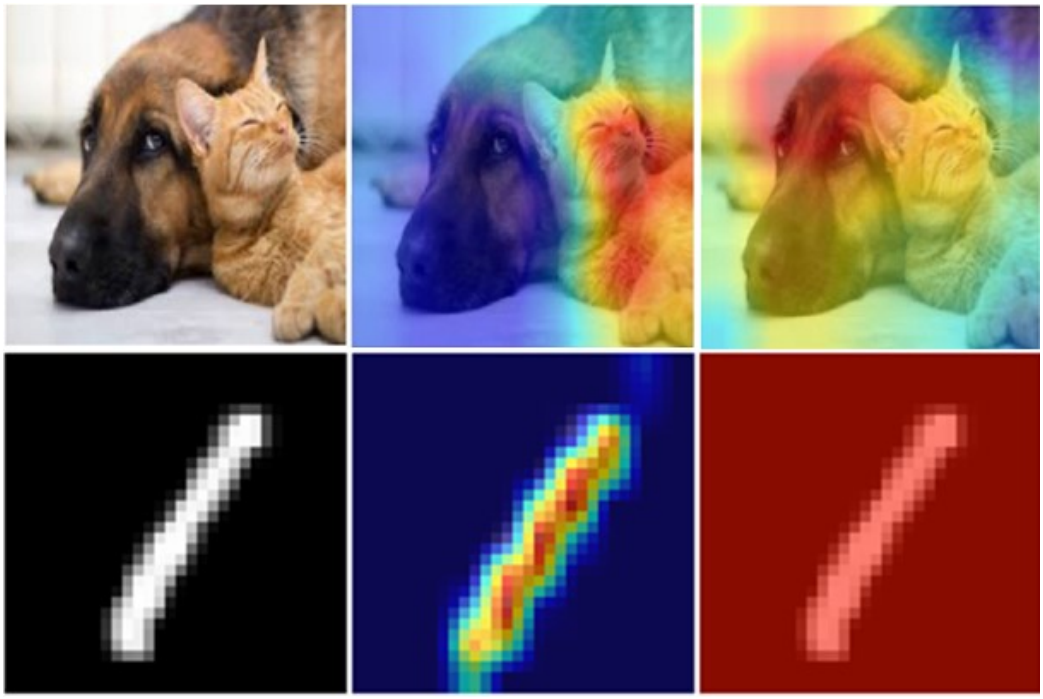

Figure 9: Sanity checks. Rows 1 and 2 present DIX results for the *parameter randomization* and *data randomization* tests w.r.t. the "tabby cat" (ImageNet) and "one" (MNIST) classes, using ResNet50 and LeNet-5, respectively. Left to right: Row 1: Original image, explanation map produced by DIX and the trained model, explanation map produced by DIX and untrained model (model's weights are randomly initialized without further training). Row 2: Original image, explanation map produced by DIX and a model trained with the ground truth labels, explanation map produced by DIX and a model trained with random labels.

7 Conclusion

We presented the Deep Integrated Explanations (DIX) method for producing explanations for vision models. The uniqueness of DIX lies in its versatility together with the accumulation of maps originating from multiple layers, encompassing interpolated network representations along with their corresponding gradients. We demonstrated the applicability of DIX for explaining CNN and ViT models, where it is shown to outperform state-of-the-art explanation methods across multiple tasks, datasets, network architectures, and metrics. In order to substantiate the efficacy of the specific configuration choices inherent to DIX, an ablation study was systematically conducted. The benefits of employing multiple layers in the interpolation process, the aggregation of distinct layer maps through summation, and the combination of the representation term with its corresponding gradient in the generation of the final explanation map were prominently observed. Finally, we validated DIX as a machinery for generating faithful explanation maps via an extensive set of sanity tests.

In the future, we plan to investigate novel methodologies that build upon the foundation of DIX, incorporating a more complex integration approach that encompasses the features and gradients of the models, to produce a better explanation. Furthermore, we plan to explore further architectures and application domains such as NLP and recommender systems.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. 1, 12

[3] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. 1, 12

[4] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 966–11 976, 2022. 1, 12

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 1, 11, 12

[6] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009. 1

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. 1

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020, pp. 213–229. 1

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017. 1

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833. 1, 6

[11] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013. 1, 6

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626. 1, 6, 9, 14

[13] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 782–791. 1, 7, 12, 13, 14, 16

[14] O. Barkan, E. Hauon, A. Caciularu, O. Katz, I. Malkiel, O. Armstrong, and N. Koenigstein, "Grad-sam: Explaining transformers via gradient self-attention maps," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2882–2887. 1

[15] I. Malkiel, D. Ginzburg, O. Barkan, A. Caciularu, J. Weill, and N. Koenigstein, "Interpreting bert-based text similarity via activation and saliency maps," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3259–3268. 1

[16] O. Barkan, Y. Fuchs, A. Caciularu, and N. Koenigstein, "Explainable recommendations via attentive multi-persona collaborative filtering," in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 468–473. 1

[17] O. Barkan, T. Shaked, Y. Fuchs, and N. Koenigstein, "Modeling users' heterogeneous taste with diversified attentive user profiles," *User Modeling and User-Adapted Interaction*, pp. 1–31, 2023. 1

[18] K. Gaiger, O. Barkan, S. Tsipory-Samuel, and N. Koenigstein, "Not all memories created equal: Dynamic user representations for collaborative filtering," *IEEE Access*, vol. 11, pp. 34 746–34 763, 2023. 1

[19] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014. 1

[20] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 839–847. 1, 6, 9, 13, 14, 15

[21] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021. 1, 6, 13, 14, 15

[22] O. Barkan, O. Armstrong, A. Hertz, A. Caciularu, O. Katz, I. Malkiel, and N. Koenigstein, "Gam: Explainable visual similarity and classification via gradient activation maps," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 68–77. 1, 6, 9

[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 2017, pp. 3319–3328. 1, 6, 9, 14, 15

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 1, 7

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 1

[26] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 397–406. 1, 7, 13, 14, 16

[27] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems*, 2018, pp. 9505–9515. 2, 16, 25

[28] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2950–2958. 6

[29] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437. 6

[30] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, 2017, pp. 6970–6979. 6

[31] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 233–255, 2016. 6

[32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929. 6

[33] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE transactions on pattern analysis and machine intelligence*, 2018. 6

[34] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *ArXiv*, vol. abs/1605.01713, 2016. 6

[35] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," in *NeurIPS*, 2019. 6, 14, 15

[36] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017. 6

[37] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li, "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns," *ArXiv*, vol. abs/2008.02312, 2020. 6, 14, 15

[38] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119, 2020. 6

[39] S. S. Desai and H. G. Ramaswamy, "Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization," *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 972–980, 2020. 6, 14, 15

[40] H. Jung and Y. Oh, "Towards better explanations of class activation mapping," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1316–1324, 2021. 6, 14, 15

[41] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *ICML*, 2017. 6, 15

[42] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017. 6, 15

[43] S. Xu, S. Venugopalan, and M. Sundararajan, "Attribution in scale and space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9680–9689. 6, 14, 15, 18

[44] A. Kapishnikov, S. Venugopalan, B. Avci, B. Wedin, M. Terry, and T. Bolukbasi, "Guided integrated gradients: An adaptive path method for removing noise," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5050–5058. 6, 14, 15, 18

[45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229. 7

[46] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," *arXiv preprint arXiv:2005.00928*, 2020. 7, 11

[47] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015. 7

[48] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5797–5808. 7, 14

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 10

[50] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018. 12

[51] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry, "Xrai: Better attributions through regions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4948–4957. 13, 18

[52] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25. 13

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Computer Vision and Pattern Recognition (CVPR)*, 2009. 14, 25

[54] M. Guillaumin, D. Küttel, and V. Ferrari, "Imagenet auto-annotation with segmentation propagation," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 328–348, 2014. 14

[55] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014, cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. [Online]. Available: http://arxiv.org/abs/1405.0312 14

[56] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009. 14

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015. 16, 25

[58] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 28