
Hidden in Plain Sight – Class Competition Focuses Attribution Maps

Nils Philipp Walter¹ Jilles Vreeken¹ Jonas Fischer²

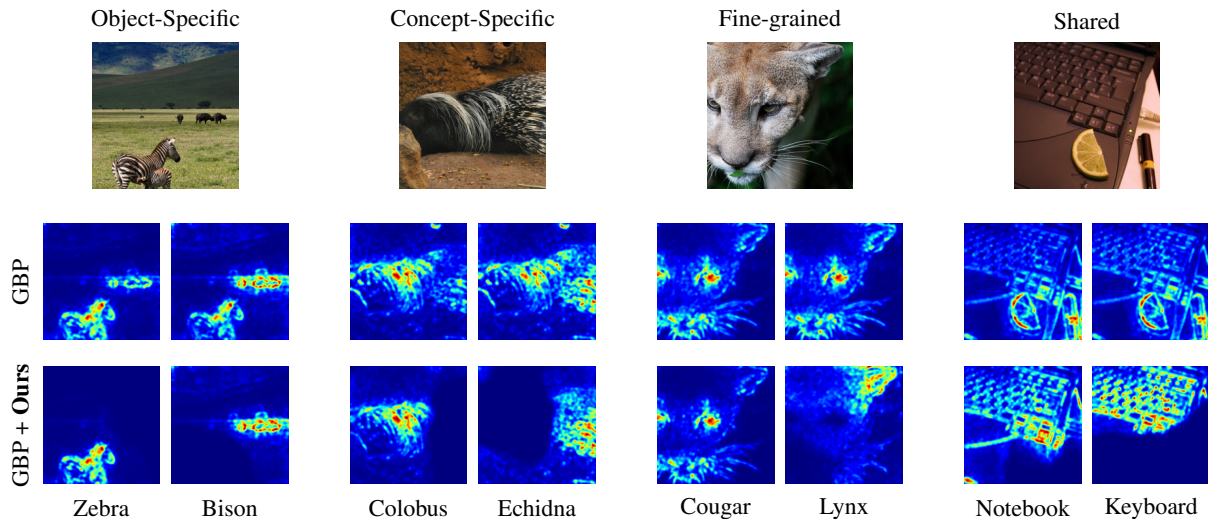


Figure 1. Attributions on ImageNet. Attributions computed as distributions across classes are: **object-specific** – visually ground correct target objects, **concept-specific**, identifying features that are relevant on a by-part-basis, and **fine-grained**, yielding features that distinguish closely related classes, while at the same time not obscuring features that are **shared** between closely related classes. In contrast, the standard approach of computing attributions on the logit of the predicted class does not reveal any of these properties.

Abstract

Attribution methods reveal which input features a neural network uses for a prediction, adding transparency to their decisions. A common problem is that these attributions seem unspecific, highlighting both important and irrelevant features. We revisit the common attribution pipeline and observe that using logits as attribution target is a main cause of this phenomenon. We show that the solution is in plain sight: considering distributions of attributions over multiple classes using existing attribution methods yields specific and fine-grained attributions. On common benchmarks, including the grid-pointing game and randomization-based sanity checks, this improves the ability of 18 attribution methods across 7 architectures up to $2\times$, agnostic to model architecture.

¹CISPA Helmholtz Center for Information Security, Saarbrücken, Deutschland ²Max Planck Institute for Informatics, Saarbrücken, Deutschland. Correspondence to: Nils Philipp Walter <nils.walter@cispa.de>.

Preprint.

1. Introduction

Neural networks are widely used for decision-making but remain opaque. This is especially problematic in high-stakes settings such as medical imaging (Borys et al., 2023), but also in a more general context, motivating the growing need for transparent explanations. A common approach to explain and understand the prediction is to highlight which features in the input, such as regions in the input, drive a prediction; such methods are termed attribution methods as they attribute an importance score to input features.

Such explanations, however, have shortcomings. While some of the attributed features appear sensible, the whole attribution seems overcomplete, making it difficult to determine which features are relevant for *discriminating* between classes (Rao et al., 2022). As illustrated in the top row of Figure 1, standard methods (e.g., GUIDEDBACKPROP) show little difference when attributing features to different classes. They tend to highlight all salient objects (e.g., both Zebra and Bison) regardless of the target. Hence, these methods currently lack the ability to differentiate between features that are distinct and important for a class and those that are just loosely associated with it.

We argue that this lack of focus is not as much a problem of the attribution methods, but rather stems from *how* we consider their output. That is, attribution is typically computed on the logit of a target class. However, looking at a logit in isolation discards the model-inherent discriminative mechanism of the subsequent softmax layer, which, for the final prediction, weighs the evidence for one class against others. Attributing to the softmax output is usually ineffective because the gradient vanishes for confident predictions, and the resulting maps become uninformative.

Here, we propose a refinement that *reintroduces* the competitive nature of the softmax without suffering from saturation to *any* existing attribution method. Instead of considering the logits in isolation, we compute the *distributions of attributions over multiple classes* (see Figure 2). This can be seen as a lens that focuses attributions by turning single-class explanations into multi-class attribution distributions, hence we call the refinement **Attribution Lens (AL)**. By analyzing how the attribution for the target class relates to conflicting classes, we unlock properties that standard logit attribution miss. Specifically, as shown in Figure 1, the resulting attributions are: (i) **object-specific**, visually grounding the correct target object (e.g., separating Zebra from Bison) (ii) **concept-specific**, identifying features relevant on a by-part basis (e.g. fur vs. spikes); (iii) **fine-grained**, yielding features that distinguish closely related classes (e.g., the ears of a Lynx vs. Cougar); and (iv) **shared**, properly identifying features common between classes (e.g., between a notebook and a keyboard) without obscuring them.

These properties are also reflected in results on established attribution benchmarks, including the grid-pointing game and a part-annotated multi-object dataset (Rao et al., 2022; Lin et al., 2014) (Table 1, App. Table 3,4), insertion tests (Table 2, App. Table 12), and randomization-based sanity checks (Adebayo et al., 2018) (Figure 6, App. Figure 9-15). On these benchmarks we show that **Attribution Lens** refines existing attribution methods while remaining agnostic to model architecture, improving benchmark metrics across 18 attribution methods and 7 architectures by up to $\sim 2\times$.

2. Related Work

In post-hoc explainability, there exist three main approaches for discovering prediction-relevant input features. Perturbation techniques probe model behavior by systematically modifying inputs, for instance by masking or deleting regions and measuring the resulting change in the model output (Petsiuk et al., 2018; Fong et al., 2019; Lundberg & Lee, 2017), and are hence computationally expensive. Approximation techniques (Ribeiro et al., 2016; Parekh et al., 2021) create interpretable surrogate models to mimic complex networks, but without guarantees that the surrogate reflects how the original model arrives at its decision.

Activation- and gradient-based attribution methods aim to balance efficiency and fidelity by considering the computation graph of the network. Well-known examples include Input \times Gradient (Simonyan et al., 2014), Integrated Gradients (Sundararajan et al., 2017; Zhuo & Ge, 2024), and GBP (Springenberg et al., 2015), which are all based on gradients through the network, and GradCAM (Selvaraju et al., 2017) and LayerCAM (Jiang et al., 2021) which upsample feature maps while taking class information into account.

Layer-wise Relevance Propagation (LRP, Bach et al., 2015) considers the flow of activation values across the network under a conservation property, which requires architecture-specific adaptations (Otsuki et al., 2024; Chefer et al., 2021). Similarly, DeepLift (Shrikumar et al., 2017) uses reference activations to determine neuron importance through custom backpropagation procedures. For transformers, recent approaches propose modifications of attention roll-out, which reflects the propagation of information through the layers by multiplying each of their transition matrices, including Bi-attn (Chen et al., 2023), T-attn (Yuan et al., 2021), and InFlow (Walker et al., 2025).

Because of their widespread use, benchmarking attribution methods in computer vision has been of growing interest. Ancona et al. (2018) study attribution sensitivity and formally proved equivalence between approaches under specific assumptions, whereas Rao et al. (2022) systematically studied how faithful attributions are to an underlying prediction using the grid-pointing game. Insertion ablations (Kapishnikov et al., 2019) instead study the effect of insertion and deletion of attributed pixels on downstream performance as a proxy for attribution quality. Adebayo et al. (2018) evaluate attribution faithfulness based on stability of explanations with randomization of network components, which was later critically revisited (Binder et al., 2023). We will use each of these metrics to study the impact of our suggested attribution approach. Orthogonally, different learning objectives have been suggested to generally improve post-hoc explanation quality such as attributions (Gairola et al., 2025), which we later relate to our findings.

3. Rethinking Attribution Computation

Post-hoc attribution methods have been shown to perform poorly in recovering the classification-relevant information from the network (Rao et al., 2022; Böhle et al., 2022) and arguably fail network perturbation based sanity checks (Adebayo et al., 2018). Commonly, the attributions for a target class—usually the predicted class—are computed using its logit as a target, which, however, means that the attribution will ignore the information from the other logits (see Fig. 2).

Recent proposals for concept visualization in diffusion models, such as ConceptAttention (Helbling et al., 2025), owe

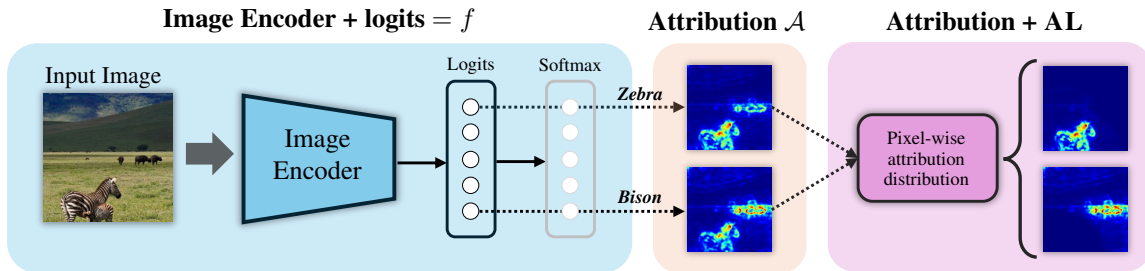


Figure 2. Reconsidering how to apply attributions. The standard approach for softmax classifiers computes attribution maps with respect to a single class logit, effectively bypassing the softmax (because of vanishing gradients). This can yield diffuse and partially redundant attributions across classes (middle). We instead compute per-class attributions and convert them into pixel-wise attribution distributions by applying a softmax over classes at each pixel (right), producing explanations that better reflect the decision process of the model.

their success to considering *multiple* concepts at once. Then, for each spatial location in the image, concept attention scores are normalized *across all concepts*, thus determining which concept the image location is most associated with.

Similarly, in classification, the final prediction is based on all logits, with softmax contrasting the logits between classes, so it is far-fetched to expect attributions to recover prediction-relevant features from a single logit alone. We propose to reconsider this common paradigm and propose to compute attributions for logits of multiple classes, appropriately normalize them, and compute *distributions over attributions* at each spatial location, similar to how a classification head computes an output distribution over multiple logits. This approach is grounded in a formal argument, which we discuss next, showing that existing attribution methods *can retrieve class-specific information* when appropriately normalized across classes.

3.1. Notation

We consider an input $x \in \mathcal{I}$, where here $\mathcal{I} = \mathbb{R}^{H \times W \times d}$ is typically an image of height H , width W , and d channels. We describe a classification model as a function $f : \mathcal{I} \rightarrow \mathbb{R}^C$, predicting C classes $\mathcal{C} = \{1, \dots, C\}$. The final class label is usually obtained as argmax over $f(x)$. We consider attributions as functions $\mathcal{A} : \mathcal{I} \times f \times \mathcal{C} \rightarrow \mathcal{I}'$ that for an input, a model, and optionally a target class provide an explanation of a similar shape as the input. For images, we typically aggregate attributions across the channel dimension. The attribution method assigns each input feature x_i a score indicating its contribution in the prediction of f for that specific class. This broad definition covers all attribution methods discussed in the related work section, examples are Input \times Gradient (IxG) as $\mathcal{A}_{\text{IxG}}(x, f, c) = x \odot \frac{\partial f_c}{\partial x}$, or GradCAM as $\mathcal{A}_{\text{GradCAM}}(x, f, c) = \text{ReLU}(\sum_k \alpha_c^k a^k)$, where $\alpha_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial f_c}{\partial a_{ij}^k}$ are the importance weights computed by global average pooling of the gradients. Although we focus on images here, the following generalizes without modification to other domains such as language.

3.2. Focusing Attribution Methods

Using a single logit f_c is non-contrastive by construction, since it does not take competing classes into account. A more principled alternative is to consider the softmax probability $p_c = \text{softmax}(f(x))_c$, which inherently contrasts between class logits. However, the gradient of p_c , which would provide the attribution signal for most attribution methods, has an important drawback. Let z_c denote the logit $f_c(x)$. The gradient of the softmax probability p_c is

$$\nabla_x p_c = p_c \left(\nabla_x z_c - \sum_{c'=1}^c p_{c'} \nabla_x z_{c'} \right). \quad (1)$$

Although the subtractive term $\sum p_{c'} \nabla_x z_{c'}$ appears to provide the necessary contrast, it is negligible in practice. In the high-confidence regime where $p_c \rightarrow 1$, the weighted sum of gradients converges to the gradient of the class with the highest logit, $\nabla_x z_c$. This causes Eq. 1 to approach zero. The resulting gradient thus fails to attribute importance to the very features that drive a confident prediction.

To build a robust contrastive attribution, we must preserve the model-inherent competition between logits while avoiding this self-canceling behavior. Our core idea is to move the contrastive mechanism from the output layer of the model, which operates on saturated probabilities, directly into the attribution maps themselves, computed for the logits. We accomplish this by a softmax over *attributions of logits* at each input location (pixel) instead of output logits.

Assume a gradient-based attribution $\mathcal{A}_c = \nabla_x z_c$ for class c . For a chosen set of classes $\mathcal{C}' \subseteq \mathcal{C}$ of size $C' = |\mathcal{C}'|$, we compute the base attribution map \mathcal{A}_c for each class $c \in \mathcal{C}'$. To create contrast, we then apply a softmax to each input feature attribution, here at each pixel (i, j) in an image,

$$\mathcal{A}_c^{\text{soft}}[i, j] = \frac{\exp(\mathcal{A}_c[i, j]/t)}{\sum_{c' \in \mathcal{C}'} \exp(\mathcal{A}_{c'}[i, j]/t)} \quad (2)$$

where t is a temperature to amplify the contrast. This yields a *distribution of attribution over classes* at each input fea-

ture $\sum_{c \in \mathcal{C}'} \mathcal{A}_c^{\text{soft}}[i, j] = 1$. These local class probabilities express how dominant each class is in each spatial location. One might now attempt to directly mimic Eq. (1) by replacing the global softmax weights p_c with the local $\mathcal{A}_c^{\text{soft}}(i, j)$, resulting in an attribution of the form:

$$\mathcal{A}_c[i, j] - \sum_{c'} \mathcal{A}_{c'}^{\text{soft}}[i, j] \mathcal{A}_{c'}[i, j]. \quad (3)$$

However, this naïve substitution reintroduces the vanishing behavior. The sum includes the self-term $\mathcal{A}_c^{\text{soft}}[i, j] \mathcal{A}_c[i, j]$, so when class c' dominates a pixel, i.e., $\mathcal{A}_{c'}^{\text{soft}}[i, j] \approx 1$, the full expression again tends to zero. Summing only over $c' \neq c$ is also problematic, as $\mathcal{A}_{c'}^{\text{soft}}$ and $\mathcal{A}_{c'}$ are strongly correlated, which may lead to overshooting.

Instead, we reduce attributions in proportion to how strongly other classes (not the target class) are influenced by a pixel through f . Rather than altering gradients for all classes, we discount the target attribution by the fraction of its evidence that is shared with non-target classes, which gives

$$\mathcal{A}_c[i, j] - \sum_{c' \neq c} \mathcal{A}_{c'}^{\text{soft}}[i, j] \mathcal{A}_c[i, j] \quad (4)$$

$$= \mathcal{A}_c[i, j] \left(1 - \sum_{c' \neq c} \mathcal{A}_{c'}^{\text{soft}}[i, j] \right) = \mathcal{A}_c[i, j] \cdot \mathcal{A}_c^{\text{soft}}[i, j]. \quad (5)$$

Thus, each pixel’s attribution is the original attribution scaled, either up or down, by its class probability $\mathcal{A}_c^{\text{soft}}[i, j]$.

3.3. From Gradients to General Attributions

Most attribution methods can be understood as functions of the gradient. They either use it directly (Input×Gradient), pool it spatially (Grad-CAM), or transform it through propagation rules (Guided Backpropagation, DeepLIFT, etc.). One way to extend our derivation would be to modify each method individually and insert the contrastive reweighting at the gradient level. However, such an approach would be cumbersome and method-specific.

We instead propose a plug-and-play refinement that operates directly on saliency maps, which we call Attribution Lens (AL, for short) as it functions as a lens that focuses on the evidence most important for the model (see Figure 2). For a subset \mathcal{C}' of classes, we compute the class-wise attributions $\mathcal{A}(x, f, c)$ and normalize them at each pixel using the spatial softmax of Eq. (2). We denote the resulting distribution by

$$\mathcal{A}_c^{\text{soft}}[i, j] := \frac{\exp(\mathcal{A}(x, f, c)[i, j]/t)}{\sum_{c' \in \mathcal{C}'} \exp(\mathcal{A}(x, f, c')[i, j]/t)}, \quad (6)$$

which expresses the relative dominance of class c at location (i, j) . To increase robustness, we average $\mathcal{A}_c^{\text{soft}}[i, j]$ over

multiple temperatures t , producing smoother distributions that capture contrast at different granularities. We then define AL to refine \mathcal{A}_c of the target class c as

$$\text{AL}(x, f, c) = \mathcal{A}(x, f, c) \odot \mathcal{A}_c^{\text{soft}} \odot \mathbb{1}_{\mathcal{A}_c^{\text{soft}} > \frac{1}{C'}}, \quad (7)$$

where \odot denotes element-wise multiplication. We mask out pixels where the target class is no more than chance among the comparison classes ($\mathcal{A}_c^{\text{soft}} \leq 1/C'$), retaining only locations where c is above-chance *within that set*. In practice, we average over $1/t \in \{1, 5, 100\}$ to stabilize the class competition. Extensive ablations on t in Appendix D.3 show that results are comparable across varying t .

Since AL operates solely on the outputs of attribution methods, it is model-agnostic and can be applied to any model and attribution function that satisfy the signature defined above. In Figure 3, we demonstrate this plug-and-play behavior across different models and attribution pairings.

3.4. Selecting the Set of Classes

One design choice of AL is the set of classes \mathcal{C}' used to compute the refinement. We explore three approaches, each offering distinct advantages depending on the specific goals and application context.

Predefined class sets. When meaningful contrasts are dictated by the task or domain, one can fix \mathcal{C}' a priori. Examples include quadrant classes in grid-pointing games or clinically relevant disease subtypes in medical imaging. This yields application-aligned attributions, but requires prior knowledge that may not be available.

Top- k most probable classes. A model-driven option is to choose the k highest-scoring classes for the input (and, optionally, also include the lowest-scoring class). Contrasting these classes require only mild computational overhead and emphasizes evidence that separates the most plausible alternatives, while also revealing features shared among closely related high-probability classes.

Best-vs-worst classes. The third approach compares the highest-probability against lowest-probability class $\mathcal{C}' = \{c_{\max}, c_{\min}\}$ with $c_{\max} = \arg \max_c f_c(x)$ and $c_{\min} = \arg \min_c f_c(x)$. Such extreme can reveal the most distinctive characteristics of the input as interpreted by the model, by showing which features are most critical for pushing the model toward or away from certain classifications and is very efficient to compute.

We use predefined sets for localization and deletion/insertion tests, and a top-2 strategy for randomization tests.

3.5. Computational Complexity

We need to compute \mathcal{C}' attribution maps, each with cost A ; hence, the runtime is $O(C'A)$. The refinement step

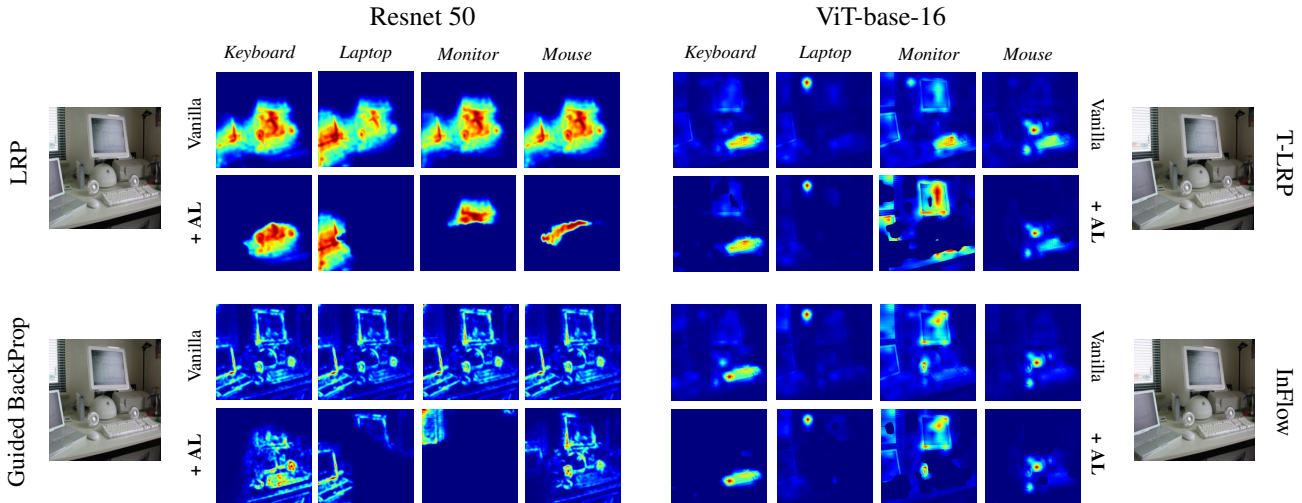


Figure 3. AL works across architectures and methods. For each baseline (top), our refinement (bottom) sharpens class-specific regions (keyboard, laptop, monitor, mouse). In ResNet-50, the effect is strongest, revealing the class-specific information clearly. In ViT-base-16, attributions already cover relevant areas but remain diffuse; AL reduces this blur and highlights the important regions cleanly.

applies softmax, weights by $\mathcal{A}^{\text{soft}}$, and thresholds, and is parallelizable making its cost negligible compared to the attribution maps. In practice, C' is small (e.g., only a few top-scoring classes), and for gradient-based methods much of the computation can be reused across classes by caching the backward path since only the final layer changes, further

4. Experiments

We empirically evaluate our attribution refinement AL across 13 gradient-based and 5 perturbation-based attribution methods on three benchmark settings. We assess (i) localization performance, (ii) insertion & deletion tests, and (iii) randomization-based sanity checks to verify robustness.

To assess localization ability, we consider the validation set of ImageNet (Russakovsky et al., 2015) and PartImageNet (He et al., 2022) to generate images for the Grid Pointing Game introduced by Rao et al. (2022), as well on the MSCOCO dataset (Lin et al., 2014). We assess the quality of attributions by measuring how well these match annotated bounding boxes and segmentation masks.

For insertion & deletion tests, we quantitatively evaluate attributions using perturbation-based metrics (Petsiuk et al., 2018), which assess whether highly attributed pixels are truly relevant to the model’s prediction. Specifically, we employ the insertion (deletion) protocol, which progressively adds (removes) the most-attributed pixels and measures the resulting change in model confidence.

To check whether attribution methods reflect what the model has learned, we run cascading randomization tests (Adebayo et al., 2018). We progressively randomize model parameters (from later to earlier layers) and verify that the resulting

attributions degrade accordingly.

We consider different architectures, including ResNet-50 (He et al., 2016), Vision Transformer B/16 (ViT) (Dosovitskiy et al., 2020), and provide further results for DenseNet-121 (Huang et al., 2017), WideResNet-50-2 (Zagoruyko & Komodakis, 2016), ConvNeXt (Liu et al., 2022), Vision Transformer B/8 and Vision Transformer B/16 in the Appendix. All models are pretrained on ImageNet and downloaded from PyTorch (Paszke et al., 2019).

For attribution methods, we adopt widely used approaches for CNNs Grad-CAM, Guided Backprop, IG, IxG, Guided Grad-CAM (Selvaraju et al., 2017; Springenberg et al., 2015; Sundararajan et al., 2017; Shrikumar et al., 2017; Kokhlikyan et al., 2020) and for ViTs Grad-CAM, InFlow, Grad-Rollout, Bi-Attn, T-attn, T-LRP, gradient saliency (Walker et al., 2025; Abnar & Zuidema, 2020; Chen et al., 2023; Yuan et al., 2021; Chefer et al., 2021). Because transformer saliency maps are blurry, we multiply them with the input (similar to IxG) for illustration.

As perturbation-based explainability methods, we consider Shapley Values and their kernel-based approximation (Kernel SHAP) (Lundberg & Lee, 2017), LIME (Ribeiro et al., 2016), as well as simple perturbation schemes such as feature ablation and occlusion (Zeiler & Fergus, 2014). These approaches quantify feature importance by selectively modifying parts of the input and observing the resulting change in the model output. We report the results in Appendix D.2.

4.1. Localization ability

Metrics We assess attribution quality by measuring how well attributions align with the actual object regions. The

Table 1. Consistent improvement of attributions. We measure the improvement of applying method across 13 different attribution methods, considering convolutional and transformer-based architectures, using Region Attribution (RA), Intersection over Union (IoU), and F1. We observe that AL consistently improves the base method. We provide results for more architectures in App. Tab. 1 showing similar trends. We show the value that the method achieves when augmented with AL and in percent the level of improvement.

Method	Quad-ImageNet			Part-Quad-ImageNet			COCO			
	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1	
Resnet50	GradCam	0.88+25%	0.67+64%	0.79+38%	0.31+28%	0.24+112%	0.36+87%	0.18+19%	0.11+16%	0.17+12%
	GBP	0.86+144%	0.26+32%	0.41+25%	0.44+146%	0.08+43%	0.14+38%	0.19+30%	0.09+3%	0.15+2%
	Guide-GC	0.91+21%	0.34+31%	0.50+23%	0.50+24%	0.12+49%	0.21+42%	0.23+16%	0.10+8%	0.16+8%
	IxG	0.55+37%	0.20+0%	0.33+0%	0.25+47%	0.06+0%	0.11+0%	0.13+11%	0.09+0%	0.15+0%
	IG	0.56+36%	0.20+0%	0.34+0%	0.28+48%	0.06+0%	0.12+0%	0.14+11%	0.09+0%	0.15+0%
	LRP	0.88+56%	0.69+97%	0.79+55%	0.37+49%	0.22+117%	0.34+90%	0.21+20%	0.13+8%	0.20+7%
	Avg. Improvement	+53.17%	+37.33%	+23.50%	+57.00%	+53.50%	+42.83%	+17.83%	+5.83%	+4.83%
ViT-base-16	Bi-attn	0.94+31%	0.71+180%	0.82+103%	0.51+40%	0.28+309%	0.40+222%	0.30+43%	0.16+52%	0.23+42%
	GradCam	0.91+6%	0.62+16%	0.75+10%	0.58+11%	0.27+39%	0.39+32%	0.31+10%	0.15+11%	0.22+9%
	InFlow	0.86+21%	0.56+126%	0.71+78%	0.53+23%	0.20+198%	0.31+153%	0.29+20%	0.13+23%	0.20+21%
	Grad-Rollout	0.73+76%	0.53+113%	0.68+71%	0.40+94%	0.20+197%	0.30+148%	0.24+30%	0.12+19%	0.19+17%
	T-attn	0.93+32%	0.71+180%	0.82+102%	0.47+38%	0.29+321%	0.40+229%	0.29+44%	0.16+53%	0.23+43%
	T-LRP	0.77+35%	0.51+105%	0.66+65%	0.47+36%	0.20+201%	0.31+152%	0.27+17%	0.12+20%	0.19+18%
	Gradient	0.93+4%	0.57+3%	0.70+2%	0.50+8%	0.34+11%	0.47+9%	0.30+10%	0.17+2%	0.25+2%
Avg. Improvement	+29.29%	+103.29%	+61.57%	+35.71%	+182.29%	+135.00%	+26.29%	+28.57%	+25.71%	

Table 2. Improving attributions on insertion test. Augmenting the base method with AL improves the AUC for insertion tests for convolutional architectures by 8-11% and modestly improves transformer attribution methods by ~2% across architectures.

(a) CNN-based architectures

Method	ResNet50	WideResNet50-2	DenseNet121	ConvNeXT
IG	0.53+13%	0.60+11%	0.50+9%	0.30+11%
GBP	0.65+25%	0.72+24%	0.64+23%	0.26+13%
IxG	0.50+14%	0.58+14%	0.45+10%	0.25+14%
Guide-GC	0.65+3%	0.72+3%	0.61+7%	0.34+3%
GradCam	0.61+0%	0.67+0%	0.51-2%	0.31-6%
LRP	0.61+11%	0.69+5%	0.44+0%	0.00+0%
Avg. Improvement	+11.00%	+9.50%	+7.83%	+5.83%

(b) Transformer-based architectures

Method	ViT-base-8	ViT-base-16	ViT-base-32
Bi-attn	0.53+4%	0.46+2%	0.52+2%
T-attn	0.53+4%	0.46+2%	0.52+2%
InFlow	0.51+4%	0.45+2%	0.51+0%
Gradient	0.52+0%	0.44+0%	0.49-2%
Grad-Rollout	0.51+9%	0.43+7%	0.50+6%
TLRP	0.00+0%	0.00+0%	0.00+0%
Avg. Improvement	+3.50%	+2.17%	+1.33%

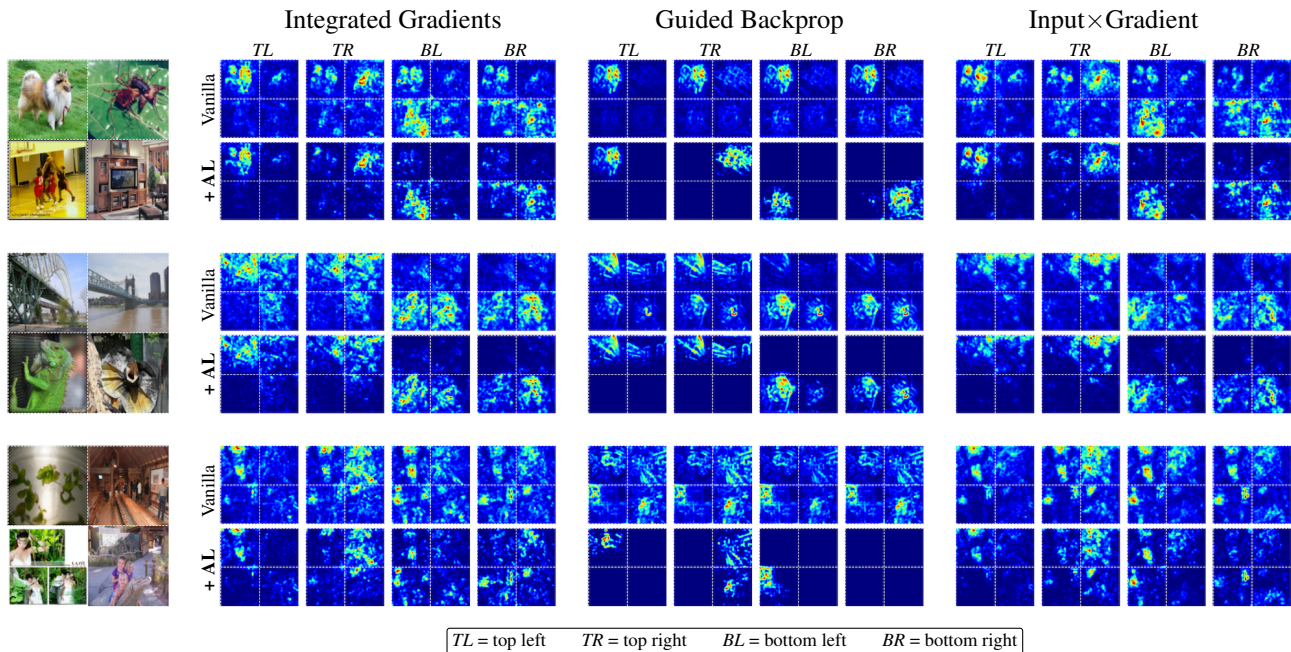


Figure 4. AL on the Grid Pointing Game. We show how AL improves the results for INTEGRATED GRADIENTS, GUIDED BACKPROP and Input x Gradient (columns) for ResNet50 on three instances (rows) for the grid pointing game. The point of the 'game' is to tell, given an input image (left) that consists of four sub-images, to tell which pixels are most relevant for the class label of a quadrant (TL for top left, TR for top right, etc.). Per image, we show the results of the vanilla (top) saliency mapping method resp. equipped with AL (bottom).

Region Attribution (RA) metric quantifies what portion of the total attribution weight falls within the target region, providing insight into attribution focus. The Intersection over Union (IoU) measures the spatial overlap between the attribution map and the ground truth region, and the F1-score score is computed between attributed and true pixels of the target object. To prevent methods from being unduly rewarded for producing diffuse attributions, we apply a Gaussian blur to the attribution maps and ensure a fair comparison across different approaches following Rao et al. (2022). For both setups, we use the ground truths for C' .

Grid Pointing For the grid-pointing game, we compile a 2×2 grid of random images from ImageNet validation set, which we call Quad-ImageNet. We also generate such grids using the PartImageNet dataset (He et al., 2022), which is a subset of ImageNet but annotated with segmentation masks. Across attribution scores and architectures, we observe that AL never degrades performance, at minimum we see that

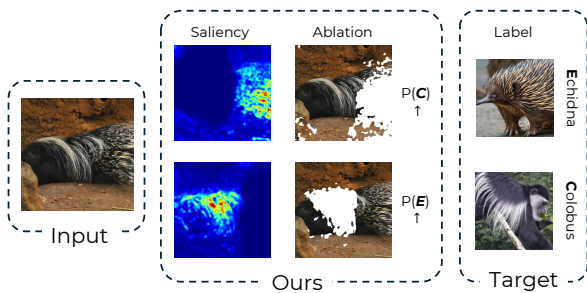


Figure 5. Qualitative example of the ablation study. To demonstrate that CL can uncover concept-specific information encoded in attribution methods, we apply it to an image of a porcupine resembling a “Colobus \times Echidna” hybrid. Using GBP on ResNet-50, we compute class-specific saliency maps for *Echidna* and *Colobus* and mask the corresponding regions: removing Echidna evidence increases the Colobus probability (and vice versa), showing that CL isolates class-discriminative cues latent in the base method.

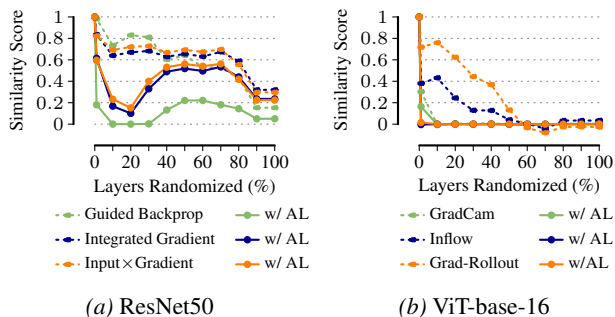


Figure 6. Sanity check by network randomization. We show similarity between attributions before and after randomization of $x\%$ of network layers for standard attribution (dashed) and when augmented with AL (solid). **Lower is better.** Randomization is from back to front of the network following Adebayo et al. (2018).

for specific methods and benchmark setups it is on par with the standard pipeline. Most of the time, we see that the **existing attribution methods refined through AL show much better localization ability** (see Tab. 1).

For ResNet50, we observe substantial gains in RA, IoU, and F1, with an average improvement of upto **+53%**, **+37%** and **+42%** across different attribution scores. For specific methods, such as Guided Backpropagation, the RA score even doubles on the Quad-Imagenet and QuadPart-ImageNet benchmark. For the ViT model, IoU and F1 scores more than double. This strong improvement can be attributed to AL filtering out uniformly unimportant regions in the noisy attribution maps produced for ViT.

Qualitatively, we also observe these improvements, now capturing both the *distinguishing* as well as *common* features of closely related classes (cf. Fig. 4). These results show that the AL pipeline enhances attribution methods in precisely localizing features most relevant to the classification.

MS-COCO For MS-COCO, we use the whole validation set. We filter objects that are smaller than 1% of the image and objects for which the model’s confidence is less than 10^{-4} . We observe similar trends, albeit more modest than on Quad-Imagenet, achieving an average improvement of **+17.83%**, **+5.83%**, and **+4.83%** on RA, IoU, and F1, respectively. Again we observe that through AL, the attributions focus on more distinctive features rather than entire object regions. COCO’s natural images contain multiple objects with complex backgrounds, making precise localization more challenging, yet with AL we do improve F1 scores across regardless of attribution scoring approach on both ResNet50 and ViT, indicating better overall localization despite the more challenging context. We provide results for different convolutional and transformer-based architectures in App. Tab. 1, showing similar improvements.

4.2. Insertion & Deletion ablations

We evaluate the actionability of refined saliency maps using standard *insertion* and *deletion* tests. For *insertion*, we start from a blurred baseline and progressively reveal the top- k pixels ranked by the attribution map, tracking the target-class probability; better attributions yield faster increases. For *deletion*, we start from the original image and progressively remove the top- k pixels, where better attributions yield faster decreases. We summarize both tests by the area under the corresponding probability curves (AUC).

To obtain a controlled multi-object setting without overly diluting probability mass, as compared to four-image grids, we construct two-image composites. Using PartImageNet segmentation masks, we retain images where at least 80% of object pixels lie within the central 50% of the image width, crop this region, and concatenate two such crops horizon-

tally to form a 224×224 input. We show an example in Appendix Figure 7. We present the results for the insertion test in Table 2 and for deletion in Appendix D.4.

We observe that adding AL substantially improves insertion AUC for CNNs, often by around 8–11 % on average, indicating much more actionable pixel rankings. Grad-CAM is the main exception; its large, blurry regions are well-suited to insertion since a big part of the image is introduced at once, so sharpening/refining them can slightly reduce insertion AUC even if the maps look cleaner. For ViTs the gains are smaller because their baseline attributions are already relatively focused, so AL mainly denoises/cleans them up rather than dramatically re-localizing evidence.

To demonstrate that AL can uncover concept-specific information encoded in attribution methods, we apply it to an image of a porcupine. Visually, the porcupine resembles a “Colobus \times Echidna” hybrid (Figure 8). We compute class-specific saliency maps for the two target classes and then mask the corresponding explanation regions. Figure 5 illustrates this procedure for AL with GBP on ResNet-50 (additional examples are provided in Figure 8).

We observe that AL highlights class-relevant concepts more clearly than the base method (Figure 1). Importantly, this also indicates that the base method already contains the relevant concept information—however, it is not directly exposed. With AL, we can extract and isolate this latent signal. As expected, removing the Echidna evidence increases the Colobus probability, and removing the Colobus evidence increases the Echidna probability. This confirms that the **attributions refined with AL identify genuinely class-discriminative cues**.

4.3. Sanity Checks

To check whether attribution methods reflect what the model has learned, we run cascading randomization tests (Adebayo et al., 2018). We progressively randomize parameters from output to input layers and track how attribution maps change as model information is destroyed. We follow the original protocol and report Spearman correlation between attributions before and after randomization (Fig. 6). We also include cosine similarity and Pearson correlation across architectures in App. Fig. 9–15, which show similar trends. Ideally, once later layers are randomized, attribution maps should contain little target-relevant information. As noted by Binder et al. (2023), these tests have limitations because they can “preserve scales of forward pass activations with high probability.” We therefore focus on the *relative change* in attributions with and without AL.

We find that attribution maps using the AL pipeline yield better results for all baseline methods and across randomization percentages. For Guided-Backprop and Input \times

Gradient, the improvement is most pronounced, as well as for randomizing the latest layers, which carry most of the conceptual meaning for the classification. Intriguingly, for ViT models, we observe that after randomization at any point in the network the similarity score is virtually zero, meaning that **specific attribution methods taking class contrast into account can pass the sanity check**.

5. Discussion & Conclusion

Attribution methods are widely used but often criticized for failing to highlight decision-relevant features. We argue that attributing a single target logit misrepresents how classifiers make decisions, and instead proposed attribution distributions over multiple classes. Through this change, we enable existing attribution methods to capture object- and concept-specific structure, revealing both class-discriminative and shared features that single-logit attributions overlook. Crucially, we show that even standard CNN attributions already encode rich class-specific signals—*hidden in plain sight*.

To quantitatively substantiate these claims, we evaluated across 18 attribution methods, 7 CNN and transformer architectures, and common attribution benchmarks, including the grid pointing game (Rao et al., 2022), sanity checks (Adebayo et al., 2018), and insertion/deletion tests. We acknowledge the interpretability literature offers many additional protocols, yet these widely adopted benchmarks provide a thorough quantitative assessment.

The reconsideration of where and how to apply attributions is method and model agnostic, training-free, and faithful to the target model in that we do not use surrogates or other, eg. generative, models that could introduce new biases. Interestingly, Gairola et al. (2025) recently found that training with binary cross-entropy loss significantly improves attributions in terms of downstream benchmarks, arguing for BCE for improved post-hoc explanations. Our findings provide a reason why this is the case, as BCE incentivizes the network to learn class-specific features, which will consequently appear in attributions even in the standard attribution pipeline looking at a logit in isolation.

Here, we compute attributions as distributions over classes using pre-computed attribution maps, which is training-free, thus maintaining the accuracy of the model, and not only reveals class-specific but also features shared across classes, which BCE discourages in training, and reveals object-specific attributions in multi-object settings.

The proposed refinement improves existing attribution methods by reintroducing the competitive mechanism of the softmax into the attribution process, thereby helping to *understand the distinguishing features a model uses for prediction*. It is training-free and can be *combined with any attribution method*, independent of architecture or attribution type.

Impact Statement

In this work, we show that common attribution methods are better than commonly believed, but only if one uses them correctly. By introducing a lightweight refinement, which we call Attribution Lens (AL), we are able to extract this information. Hence, our work positively contributes to a better understanding the decision-making of neural networks and thus to ultimately a safer deployment of such approaches. Hence, we do not expect any potentially negative consequences caused by our work.

References

- Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS, 2018.
- Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*, 2015.
- Binder, A., Weber, L., Lapuschkin, S., Montavon, G., Müller, K.-R., and Samek, W. Shortcomings of top-down randomization-based sanity checks for evaluations of deep neural network explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- Böhle, M., Fritz, M., and Schiele, B. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- Borys, K., Schmitt, Y. A., Nauta, M., Seifert, C., Krämer, N., Friedrich, C. M., and Nensa, F. Explainable ai in medical imaging: An overview for clinical practitioners – beyond saliency-based xai approaches. *European Journal of Radiology*, 2023.
- Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- Chen, J., Li, X., Yu, L., Dou, D., and Xiong, H. Beyond intuition: Rethinking token attributions inside transformers. *Transactions on Machine Learning Research*, 2023.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., and Houtsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020.
- Fong, R., Patrick, M., and Vedaldi, A. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.
- Gairola, S., Böhle, M., Locatello, F., and Schiele, B. How to probe: Simple yet effective techniques for improving post-hoc explanations. In *The Thirteenth International Conference on Learning Representations*, 2025.
- He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.-N., Liu, S., Yang, C., Yu, Q., and Yuille, A. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, 2022.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- Helbling, A., Meral, T. H. S., Hoover, B., Yanardag, P., and Chau, D. H. Conceptattention: Diffusion transformers learn highly interpretable features. In *Proceedings of the International Conference on Machine Learning*, 2025.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M., and Wei, Y. Layercam: Exploring hierarchical class activation maps for localization. *IEEE transactions on image processing*, 2021.
- Kapishnikov, A., Bolukbasi, T., Viégas, F., and Terry, M. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.
- Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv*, 2020.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014.
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.
- Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 2017.
- Otsuki, S., Iida, T., Doublet, F., Hirakawa, T., Yamashita, T., Fujiyoshi, H., and Sugiura, K. Layer-wise relevance propagation with conservation property for resnet. In *European Conference on Computer Vision*, 2024.
- Parekh, J., Mozharovskiy, P., and d’Alché Buc, F. A framework to learn with interpretation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference*, 2018.

- Rao, S., Böhle, M., and Schiele, B. Towards better understanding attribution methods. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2022.
- Ribeiro, M. T., Singh, S., and Guestrin, C. Why should i trust you?: Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, 2015.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning*, 2017.
- Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshops at International Conference on Learning Representations*, 2014.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *In workshops of International Conference on Learning Representations*, 2015.
- Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- Walker, C., Ahmed, M. R., Jha, S. K., and Ewetz, R. Explaining vits using information flow. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.
- Yuan, T., Li, X., Xiong, H., Cao, H., and Dou, D. Explaining information flow inside vision transformers using markov chain. In *XAI 4 Debugging Workshop @ NEURIPS 2021*, 2021.
- Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 2014.
- Zhuo, Y. and Ge, Z. Ig2: Integrated gradient on iterative gradient path for feature attribution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

A. Method

A.1. Selecting the Set of Classes

Having defined our class-relevant attribution operator \mathcal{C}_H , an important consideration is the selection of the set of classes K used for calculation. We explore three approaches for class selection, each offering distinct advantages depending on the specific analysis goals and application context.

Predefined Class Sets. The canonical approach is to use a predefined set of classes K that are of particular interest. This is especially useful in contexts where specific class comparisons have natural interpretations. For example, in a grid-pointing game where users must identify the quadrant containing a particular object, the four quadrant classes directly correspond to the task structure. Similarly, in medical applications, contrasting disease subtypes can highlight discriminative features that aid differential diagnosis. This approach ensures that the resulting attributions focus on distinctions that are meaningful to the particular application domain. However, this approach requires specific knowledge about the task, which is often not available. The following approaches are data- and model-driven and, hence, do not require prior knowledge to select classes.

Top- k Most Probable Classes. A model dependent approach to class selection involves choosing the k classes with highest predicted probabilities and the class with the lowest probability for a given input. This approach is particularly effective for highlighting the features that distinguish between the most plausible classifications for a given input, but also reveal information that is shared between highly related classes that are likely among the highest probabilities. As these classes represent the top candidates for the final classification, contrasting their attribution maps reveals the most decision-relevant features.

Best-vs-Worst Classes. The third approach compares the highest-probability class against the lowest-probability class: $K = \{c_{\max}, c_{\min}\}$ where $c_{\max} = \arg \max_c S_c(x)$ and $c_{\min} = \arg \min_c S_c(x)$. Such extreme can surprisingly reveal the most distinctive characteristics of the input as interpreted by the model, by showing which features are most critical for pushing the model toward or away from certain classifications.

B. Evaluation Metrics

In our experimental setup, we evaluate attribution methods across several metrics to assess their efficacy in highlighting relevant features for model predictions. We define an input as a vector $x \in \mathbb{R}^d$, and a model as a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^C$, where C is the number of classes in the classification problem. The final classification is performed via an argmax over $f(x)$. An explanation method provides an explanation map $\mathcal{A} : \mathbb{R}^d \times S \times \{1, \dots, C\} \rightarrow \mathbb{R}^d$ that maps an input, a model, and optionally a target class to an attribution map of the same shape as the input.

B.1. Localization metrics

We evaluate attribution methods using two datasets: a Grid Pointing Game based on ImageNet and COCO dataset with segmentation masks. For both evaluations, we apply the same set of metrics, treating both bounding boxes and segmentation masks as regions of interest R in the image. We match the region of interest with the correct attribution map \mathcal{A}_c i.e. for the first quadrant we also take the first attribution map. We only take the positive part of \mathcal{A}_c . Before evaluation, we apply a Gaussian blur with a kernel size of 11×11 to the attribution maps

$$\tilde{\mathcal{A}}_c = \mathcal{G}_\sigma * \mathcal{A}_c,$$

where \mathcal{G}_σ is a Gaussian kernel with standard deviation σ and $*$ denotes the convolution operation. This preprocessing is common to prevent methods from being unduly rewarded for producing diffuse attribution maps. We then compute the following metrics:

B.1.1. REGION ATTRIBUTION

We quantify what fraction of the total positive attribution falls within the region of interest by

$$\text{RA} = \frac{\sum_{i \in R} \tilde{\mathcal{A}}_c(i)}{\sum_i \tilde{\mathcal{A}}_c(i)}.$$

B.1.2. INTERSECTION OVER UNION (IoU)

We compute the overlap between the attribution map and the region of interest as

$$\text{IoU} = \frac{|\tilde{\mathcal{A}}_c \cap R|}{|\tilde{\mathcal{A}}_c \cup R|}.$$

B.1.3. PRECISION AND RECALL

To calculate precision and recall, we use the common intersection-based formulas

$$\text{Precision} = \frac{|\tilde{\mathcal{A}}_c \cap R|}{|\tilde{\mathcal{A}}_c|}, \quad \text{Recall} = \frac{|\tilde{\mathcal{A}}_c \cap R|}{|R|}.$$

B.1.4. F1 SCORE

To calculate F1, we make use of the previously defined precision and recall metrics, calculating

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

C. LLM Use

In this work, we used GPT-5 for both writing and coding support. On the writing side, it assisted with editing and condensing text to improve clarity. For coding, GPT-5 was used for debugging, providing autocomplete suggestions in VS Code, and generating code for LaTeX figures.

D. Additional Result

D.1. Localization

We provide additional results for all the architectures mentioned in the Experiments in Table 3. The trend remains the same across architectures and methods; if they are augmented using AL they improve the localization metrics and trade-off recall. Additionally we provide plots similar to Figure 4 for all these architectures in Figure 16-22.

D.2. Perturbation based methods

We compute perturbation-based attribution maps with and provide additional results for all architectures used in the Experiments in Table 3. For Shapley Value Sampling, KernelSHAP, Feature Ablation, and LIME we define “features” as SLIC superpixels (fixed to 100 segments with compactness 10) and attribute the target class by perturbing superpixels against a common baseline. Occlusion instead perturbs fixed square patches: we occlude 1515 regions (jointly over all channels) placed on a regular grid with stride 8 pixels; overlapping occlusions are aggregated to obtain a dense heatmap. Unless stated otherwise, we keep method hyperparameters fixed across models (e.g., $n_{samples} = 64$ for Shapley/KernelSHAP and $n_{samples} = 1000$ for LIME).

Across both Quad-ImageNet and Part-Quad-ImageNet we observe a consistent and strict improvement when enabling VAR, indicating better localization quality under the same evaluation protocol. On COCO, VAR increases region accuracy (RA), but degrades F1 and IoU; this discrepancy is primarily driven by the superpixel-based perturbation setup. In particular, the superpixels (and resulting patches) often fail to align with full object extents in cluttered multi-object scenes, which reduces coverage of the ground-truth regions and leads to a pronounced drop in recall—ultimately hurting F1 and IoU despite improved RA.

D.3. Sensitivity of AL regarding scaling parameters

In this section, we investigate the sensitivity with respect to the hyperparameter of the scaling as described in Section 3.3. In all our experiments, we use the parameters $t = \{1, t_1, t_2\}$ with $t_1 = 5$ and $t_2 = 100$. We now vary $t_1 \in \{2, 5, 10\}$ and $t_2 \in \{50, 100, 500\}$. For this experiment we focus on the standard gridpointing game as described in Section 4.1. We present the results for all architectures and attribution methods as in the original setup in Table 5 - 11.

D.4. Deletion test

We report the results for the deletion test in Table 12. In contrast to insertion tests a lower AUC is better. For CNNs, adding AL consistently reduces AUC (often 9–15% on average for ResNet/WideResNet/DenseNet), i.e., removing the top-ranked pixels identified by AL drops the target probability faster and thus better targets class-critical evidence. The main exception is Grad-CAM (and ConvNeXT overall), where coarse, high-coverage maps can behave like near “one-shot” masks in deletion, leaving little room for refinement and sometimes worsening AUC when refinement becomes more selective. For ViTs, gains are limited and sometimes slightly negative on average: many transformer attributions are relatively diffuse, so deleting their top-ranked regions removes large image areas and can look strong under deletion, whereas AL tends to denoise/localize and therefore deletes less context early.

Deletion differs from insertion because confidence can fall not only when truly class-relevant evidence is removed, but also when unrelated yet supportive context (or general image structure) is destroyed; consequently, large or blurry masks may score well by broadly degrading the input rather than precisely isolating discriminative cues.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

Method	Quad-ImageNet			Part-Quad-ImageNet			COCO			
	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1	
Resnet50	GradCam	0.88+25%	0.67+64%	0.79+38%	0.31+28%	0.24+112%	0.36+87%	0.18+19%	0.11+16%	0.17+12%
	GBP	0.86+144%	0.26+32%	0.41+25%	0.44+146%	0.08+43%	0.14+38%	0.19+30%	0.09+3%	0.15+2%
	Guide-GC	0.91+21%	0.34+31%	0.50+23%	0.50+24%	0.12+49%	0.21+42%	0.23+16%	0.10+8%	0.16+8%
	IxG	0.55+37%	0.20+0%	0.33+0%	0.25+47%	0.06+0%	0.11+0%	0.13+11%	0.09+0%	0.15+0%
	IG	0.56+36%	0.20+0%	0.34+0%	0.28+48%	0.06+0%	0.12+0%	0.14+11%	0.09+0%	0.15+0%
	LRP	0.88+56%	0.69+97%	0.79+55%	0.37+49%	0.22+117%	0.34+90%	0.21+20%	0.13+8%	0.20+7%
	Avg. Improvement	+53.17%	+44.8%	+28.2%	+57.0%	+80.25%	+64.25%	+18.33%	+6.17%	+5.33%
Wide-Resnet502	GradCam	0.88+22%	0.66+62%	0.78+37%	0.30+23%	0.24+108%	0.36+84%	0.18+14%	0.11+11%	0.17+9%
	GBP	0.89+109%	0.28+42%	0.43+32%	0.47+108%	0.09+57%	0.16+51%	0.20+24%	0.10+2%	0.16+2%
	Guide-GC	0.92+17%	0.35+32%	0.51+24%	0.51+20%	0.13+53%	0.22+46%	0.24+13%	0.10+6%	0.17+6%
	IxG	0.62+38%	0.20+0%	0.33+0%	0.27+48%	0.06+0%	0.11+0%	0.15+12%	0.10+0%	0.15+0%
	IG	0.62+37%	0.20+0%	0.34+0%	0.31+48%	0.06+0%	0.12+0%	0.15+12%	0.10+0%	0.15+0%
	LRP	0.89+49%	0.72+115%	0.82+65%	0.37+46%	0.22+132%	0.34+102%	0.22+22%	0.13+11%	0.20+9%
	Avg. Improvement	+45.33%	+41.83%	+26.33%	+48.83%	+58.33%	+47.17%	+16.17%	+5.0%	+4.33%
Densenet121	GradCam	0.60+17%	0.37+6%	0.48-2%	0.22+30%	0.15+51%	0.23+39%	0.11-11%	0.07-25%	0.11-25%
	GBP	0.85+158%	0.25+27%	0.40+21%	0.41+159%	0.08+35%	0.14+32%	0.19+36%	0.10+3%	0.15+3%
	Guide-GC	0.71+28%	0.26+12%	0.40+9%	0.37+34%	0.10+30%	0.17+26%	0.17+3%	0.08-5%	0.14-5%
	IxG	0.46+31%	0.20+0%	0.33+0%	0.20+42%	0.06+0%	0.11+0%	0.13+10%	0.09+0%	0.15+0%
	IG	0.50+34%	0.20+0%	0.34+0%	0.24+46%	0.06+0%	0.12+0%	0.14+11%	0.09+0%	0.15+0%
	LRP	0.44+24%	0.25+0%	0.40+0%	0.19+35%	0.07+0%	0.12+0%	0.15+5%	0.12+0%	0.18+0%
	Avg. Improvement	+48.67%	+7.5%	+4.67%	+57.67%	+19.33%	+16.17%	+9.0%	-4.5%	-4.5%
Convnext	GradCam	0.96+2%	0.55-7%	0.70-6%	0.48+8%	0.29+31%	0.42+24%	0.28+8%	0.15+2%	0.23+2%
	GBP	0.52+26%	0.20+0%	0.33+0%	0.19+33%	0.06+0%	0.11+0%	0.15+13%	0.09+0%	0.15+0%
	Guide-GC	0.96+1%	0.35+1%	0.52+1%	0.58+5%	0.16+2%	0.26+2%	0.31+5%	0.14+1%	0.22+1%
	IxG	0.51+27%	0.20+0%	0.33+0%	0.19+33%	0.06+0%	0.11+0%	0.15+13%	0.09+0%	0.15+0%
	IG	0.64+35%	0.21+0%	0.34+0%	0.26+49%	0.06+1%	0.12+1%	0.15+16%	0.09+0%	0.15+0%
	Avg. Improvement	+18.20%	-1.20%	-1%	+25.60%	+6.80%	+5.40%	+11.00%	+0.60%	+0.60%
	ViT-base-8	Bi-attn	0.91+48%	0.62+149%	0.76+89%	0.56+61%	0.25+272%	0.36+199%	0.29+45%	0.14+32%
GradCam		0.83+8%	0.49+18%	0.64+12%	0.61+11%	0.28+46%	0.40+36%	0.30+13%	0.14+9%	0.21+7%
InFlow		0.82+18%	0.47+89%	0.63+58%	0.59+19%	0.18+165%	0.28+131%	0.32+18%	0.12+14%	0.19+13%
Grad-Rollout		0.71+51%	0.45+80%	0.61+53%	0.48+60%	0.20+197%	0.30+147%	0.26+27%	0.12+14%	0.19+12%
T-attn		0.90+53%	0.63+152%	0.76+90%	0.51+76%	0.28+322%	0.40+230%	0.28+56%	0.14+34%	0.22+29%
LRP		0.76+25%	0.42+69%	0.58+46%	0.54+24%	0.20+195%	0.30+148%	0.28+16%	0.12+14%	0.19+13%
Gradient		0.90+7%	0.49+7%	0.64+5%	0.57+11%	0.35+20%	0.48+16%	0.31+17%	0.16+2%	0.23+1%
Avg. Improvement		+30.0%	+80.57%	+50.43%	+37.43%	+173.86%	+129.57%	+27.43%	+17.0%	+14.57%
ViT-base-16	Bi-attn	0.94+31%	0.71+180%	0.82+103%	0.51+40%	0.28+309%	0.40+222%	0.30+43%	0.16+52%	0.23+42%
	GradCam	0.91+6%	0.62+16%	0.75+10%	0.58+11%	0.27+39%	0.39+32%	0.31+10%	0.15+11%	0.22+9%
	InFlow	0.86+21%	0.56+126%	0.71+78%	0.53+23%	0.20+198%	0.31+153%	0.29+20%	0.13+23%	0.20+21%
	Grad-Rollout	0.73+76%	0.53+113%	0.68+71%	0.40+94%	0.20+197%	0.30+148%	0.24+30%	0.12+19%	0.19+17%
	T-attn	0.93+32%	0.71+180%	0.82+102%	0.47+38%	0.29+321%	0.40+229%	0.29+44%	0.16+53%	0.23+43%
	LRP	0.77+35%	0.51+105%	0.66+65%	0.47+36%	0.20+201%	0.31+152%	0.27+17%	0.12+20%	0.19+18%
	Gradient	0.93+4%	0.57+3%	0.70+2%	0.50+8%	0.34+11%	0.47+9%	0.30+10%	0.17+2%	0.25+2%
	Avg. Improvement	+29.29%	+103.29%	+61.57%	+35.71%	+182.29%	+135.0%	+24.86%	+25.71%	+21.71%
ViT-base-32	Bi-attn	0.86+71%	0.62+149%	0.75+87%	0.36+80%	0.24+263%	0.36+195%	0.21+37%	0.13+33%	0.20+27%
	GradCam	0.78+18%	0.51+50%	0.65+30%	0.41+28%	0.22+119%	0.32+91%	0.22+25%	0.13+15%	0.20+13%
	InFlow	0.78+21%	0.56+124%	0.70+75%	0.38+24%	0.19+176%	0.29+136%	0.22+19%	0.12+21%	0.19+17%
	Grad-Rollout	0.66+91%	0.51+106%	0.66+66%	0.27+112%	0.17+151%	0.27+119%	0.17+28%	0.11+13%	0.18+12%
	T-attn	0.84+70%	0.62+146%	0.74+86%	0.35+77%	0.25+267%	0.36+197%	0.20+35%	0.14+34%	0.20+27%
	LRP	0.66+49%	0.46+85%	0.61+53%	0.31+51%	0.17+147%	0.26+115%	0.20+16%	0.11+11%	0.18+9%
	Gradient	0.79+19%	0.51+19%	0.65+12%	0.36+27%	0.23+53%	0.34+42%	0.21+17%	0.13+11%	0.20+8%
	Avg. Improvement	+48.43%	+97.0%	+58.43%	+57.0%	+168.0%	+127.86%	+25.29%	+19.71%	+16.14%

Table 3. Consistent improvement of attributions. Across 11 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 get consistently improved by a wide margin.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

Method	Quad-ImageNet			Part-Quad-ImageNet			COCO			
	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1	
ResNet50	Feature Ablation	0.71+31%	0.35+33%	0.51+22%	0.40+42%	0.14+69%	0.23+57%	0.20+13%	0.11-1%	0.17+1%
	Kernel SV	0.51+23%	0.30+12%	0.46+7%	0.26+36%	0.12+44%	0.20+38%	0.16+3%	0.11-7%	0.17-4%
	LIME	0.85+24%	0.48+43%	0.64+28%	0.50+32%	0.17+71%	0.27+58%	0.27+14%	0.13+5%	0.21+6%
	Occlusion	0.69+30%	0.32+24%	0.48+17%	0.41+45%	0.11+50%	0.20+43%	0.22+14%	0.11+0%	0.18+2%
	Shapley Values	0.83+27%	0.47+52%	0.64+34%	0.49+35%	0.17+83%	0.27+67%	0.26+17%	0.13+6%	0.20+8%
	Avg. Improvement	+27.00%	+32.80%	+21.60%	+38.00%	+63.40%	+52.60%	+12.20%	+0.60%	+2.60%
WRN50-2	Feature Ablation	0.74+28%	0.36+37%	0.52+25%	0.40+35%	0.14+68%	0.23+57%	0.19+12%	0.10-1%	0.16+0%
	Kernel SV	0.53+24%	0.31+13%	0.46+9%	0.26+36%	0.12+44%	0.21+38%	0.15+4%	0.11-4%	0.17-2%
	LIME	0.87+20%	0.51+45%	0.66+29%	0.50+28%	0.18+71%	0.28+58%	0.27+14%	0.12+4%	0.20+5%
	Occlusion	0.72+28%	0.33+28%	0.49+20%	0.41+39%	0.12+49%	0.20+42%	0.19+11%	0.10-2%	0.16+0%
	Shapley Values	0.86+23%	0.51+57%	0.66+37%	0.50+29%	0.18+84%	0.28+68%	0.26+18%	0.12+8%	0.19+9%
	Avg. Improvement	+24.60%	+36.00%	+24.00%	+33.40%	+63.20%	+52.60%	+11.80%	+1.00%	+2.40%
DenseNet121	Feature Ablation	0.63+30%	0.31+22%	0.46+14%	0.37+38%	0.13+60%	0.22+50%	0.20+12%	0.11-3%	0.17-1%
	Kernel SV	0.50+22%	0.29+9%	0.44+6%	0.25+35%	0.12+41%	0.20+35%	0.17+3%	0.11-5%	0.18-3%
	LIME	0.82+25%	0.45+37%	0.61+24%	0.49+31%	0.16+67%	0.27+55%	0.28+14%	0.14+5%	0.21+6%
	Occlusion	0.65+28%	0.32+20%	0.48+14%	0.38+39%	0.12+45%	0.20+38%	0.21+13%	0.11-1%	0.18+1%
	Shapley Values	0.80+28%	0.45+45%	0.61+30%	0.48+34%	0.16+78%	0.27+63%	0.27+16%	0.13+6%	0.21+8%
	Avg. Improvement	+26.60%	+26.60%	+17.60%	+35.40%	+58.20%	+48.20%	+11.60%	+0.40%	+2.20%
ConvNeXT	Feature Ablation	0.67+29%	0.33+28%	0.48+19%	0.30+35%	0.10+38%	0.18+33%	0.18+16%	0.10-4%	0.16-1%
	Kernel SV	0.57+27%	0.32+16%	0.48+11%	0.25+42%	0.12+46%	0.20+41%	0.16+6%	0.10-4%	0.17-1%
	LIME	0.90+15%	0.52+43%	0.68+28%	0.49+21%	0.19+58%	0.30+46%	0.30+9%	0.12+4%	0.20+4%
	Occlusion	0.57+26%	0.28+14%	0.43+11%	0.21+32%	0.07+14%	0.13+12%	0.18+12%	0.10-4%	0.15-2%
	Shapley Values	0.92+17%	0.58+79%	0.73+49%	0.49+26%	0.19+111%	0.30+88%	0.27+18%	0.12+10%	0.20+11%
	Avg. Improvement	+22.80%	+36.00%	+23.60%	+31.20%	+53.40%	+44.00%	+12.20%	+0.40%	+2.20%
ViT-base-16	Feature Ablation	0.59+32%	0.29+18%	0.44+11%	0.30+46%	0.11+50%	0.19+42%	0.18+15%	0.10-3%	0.17-1%
	Kernel SV	0.53+26%	0.30+14%	0.46+9%	0.25+39%	0.12+45%	0.20+39%	0.16+5%	0.11-4%	0.18-2%
	LIME	0.88+18%	0.50+48%	0.65+31%	0.49+24%	0.17+72%	0.28+58%	0.31+12%	0.14+5%	0.22+5%
	Occlusion	0.53+31%	0.27+12%	0.41+8%	0.27+45%	0.09+27%	0.15+24%	0.19+14%	0.10-3%	0.17-1%
	Shapley Values	0.86+24%	0.49+63%	0.65+41%	0.47+32%	0.17+95%	0.27+77%	0.28+19%	0.13+9%	0.21+10%
	Avg. Improvement	+26.20%	+31.00%	+20.00%	+37.20%	+57.80%	+48.00%	+13.00%	+0.80%	+2.20%
ViT-base-8	Feature Ablation	0.57+28%	0.26+12%	0.40+7%	0.30+36%	0.10+39%	0.18+33%	0.18+9%	0.10-4%	0.16-3%
	Kernel SV	0.39+13%	0.25-2%	0.39-3%	0.16+24%	0.10+24%	0.17+21%	0.14+4%	0.11-5%	0.17-3%
	LIME	0.90+16%	0.49+44%	0.65+28%	0.51+19%	0.18+63%	0.28+52%	0.30+10%	0.13+2%	0.21+3%
	Occlusion	0.42+28%	0.21+0%	0.33-1%	0.19+42%	0.06+12%	0.12+10%	0.16+8%	0.09-8%	0.15-6%
	Shapley Values	0.88+20%	0.49+60%	0.65+39%	0.49+26%	0.17+87%	0.27+70%	0.27+16%	0.13+6%	0.20+7%
	Avg. Improvement	+21.00%	+22.80%	+14.00%	+29.40%	+45.00%	+37.20%	+9.40%	-1.80%	-0.40%
ViT-base-32	Feature Ablation	0.63+26%	0.34+22%	0.49+14%	0.29+32%	0.13+48%	0.21+40%	0.19+9%	0.11-4%	0.17-2%
	Kernel SV	0.42+17%	0.27+2%	0.41+0%	0.17+26%	0.10+26%	0.17+23%	0.15+1%	0.11-7%	0.17-4%
	LIME	0.72+26%	0.37+24%	0.53+16%	0.37+40%	0.14+58%	0.23+48%	0.26+14%	0.13+5%	0.20+6%
	Occlusion	0.62+26%	0.31+17%	0.47+12%	0.29+35%	0.11+35%	0.18+30%	0.20+10%	0.11-2%	0.18+0%
	Shapley Values	0.72+28%	0.40+35%	0.56+23%	0.37+42%	0.14+69%	0.24+57%	0.25+17%	0.13+6%	0.20+7%
	Avg. Improvement	+24.60%	+20.00%	+13.00%	+35.00%	+47.20%	+39.60%	+10.20%	-0.40%	+1.40%

Table 4. Across 5 different perturbation based attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 get consistently improved.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

ResNet50										
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	GradCam	0.88+25%	0.67+64%	0.79+39%	0.88+25%	0.67+64%	0.79+39%	0.88+25%	0.67+64%	0.79+39%
	GBP	0.85+140%	0.26+32%	0.40+25%	0.85+140%	0.26+32%	0.40+25%	0.85+140%	0.26+32%	0.40+25%
	Guide-GC	0.91+21%	0.34+31%	0.50+23%	0.91+21%	0.34+30%	0.50+23%	0.91+21%	0.34+30%	0.50+23%
	IxG	0.54+33%	0.20+0%	0.33+0%	0.54+34%	0.20+0%	0.33+0%	0.54+35%	0.20+0%	0.33+0%
	IG	0.55+33%	0.20+0%	0.34+0%	0.56+34%	0.20+0%	0.34+0%	0.56+35%	0.20+0%	0.34+0%
	LRP	0.87+54%	0.68+95%	0.79+54%	0.87+54%	0.68+94%	0.79+54%	0.87+54%	0.68+94%	0.79+54%
$t_2 = 100$	GradCam	0.88+25%	0.67+64%	0.79+38%	0.88+25%	0.67+64%	0.79+38%	0.88+25%	0.67+64%	0.79+38%
	GBP	0.86+144%	0.26+32%	0.41+25%	0.86+144%	0.26+32%	0.41+25%	0.87+145%	0.26+32%	0.40+25%
	Guide-GC	0.91+21%	0.34+31%	0.50+23%	0.91+21%	0.34+31%	0.50+23%	0.92+22%	0.34+30%	0.50+23%
	IxG	0.55+36%	0.20+0%	0.33+0%	0.55+37%	0.20+0%	0.33+0%	0.55+38%	0.20+0%	0.33+0%
	IG	0.56+35%	0.20+0%	0.34+0%	0.56+36%	0.20+0%	0.34+0%	0.57+37%	0.20+0%	0.34+0%
	LRP	0.88+56%	0.69+97%	0.79+55%	0.88+56%	0.69+97%	0.79+55%	0.88+56%	0.69+96%	0.79+55%
$t_2 = 500$	GradCam	0.88+25%	0.66+63%	0.78+38%	0.88+25%	0.66+63%	0.78+38%	0.88+25%	0.67+63%	0.78+38%
	GBP	0.89+153%	0.26+33%	0.41+26%	0.89+153%	0.26+33%	0.41+26%	0.90+153%	0.26+33%	0.41+26%
	Guide-GC	0.92+23%	0.34+31%	0.50+23%	0.92+23%	0.34+31%	0.50+23%	0.92+23%	0.34+31%	0.50+23%
	IxG	0.55+37%	0.20+0%	0.33+0%	0.55+38%	0.20+0%	0.33+0%	0.56+39%	0.20+0%	0.33+0%
	IG	0.56+35%	0.20+0%	0.34+0%	0.57+36%	0.20+0%	0.34+0%	0.57+37%	0.20+0%	0.34+0%
	LRP	0.89+58%	0.70+99%	0.80+56%	0.89+58%	0.70+99%	0.80+56%	0.89+58%	0.70+99%	0.80+56%

Table 5. Low variation between scaling parameters. Across 6 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

WideResNet50-2										
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	GradCam	0.88+21%	0.66+62%	0.79+37%	0.88+22%	0.66+62%	0.79+37%	0.88+22%	0.66+62%	0.79+37%
	GBP	0.87+106%	0.28+41%	0.43+32%	0.88+106%	0.28+41%	0.43+32%	0.88+107%	0.28+41%	0.43+32%
	Guide-GC	0.92+16%	0.35+32%	0.51+24%	0.92+16%	0.35+32%	0.51+24%	0.92+16%	0.35+33%	0.51+24%
	IxG	0.61+35%	0.20+0%	0.33+0%	0.61+36%	0.20+0%	0.33+0%	0.61+36%	0.20+0%	0.33+0%
	IG	0.60+34%	0.20+0%	0.34+0%	0.61+35%	0.20+0%	0.34+0%	0.61+36%	0.20+0%	0.34+0%
	LRP	0.88+48%	0.72+113%	0.82+64%	0.88+48%	0.71+112%	0.82+63%	0.89+48%	0.71+112%	0.82+63%
$t_2 = 100$	GradCam	0.88+22%	0.66+61%	0.78+37%	0.88+22%	0.66+62%	0.78+37%	0.88+22%	0.66+62%	0.79+37%
	GBP	0.89+109%	0.28+42%	0.43+32%	0.89+109%	0.28+42%	0.43+32%	0.89+109%	0.28+42%	0.43+32%
	Guide-GC	0.92+17%	0.35+32%	0.51+24%	0.92+17%	0.35+32%	0.51+24%	0.92+17%	0.35+33%	0.51+24%
	IxG	0.62+38%	0.20+0%	0.33+0%	0.62+38%	0.20+0%	0.33+0%	0.62+39%	0.20+0%	0.33+0%
	IG	0.61+36%	0.20+0%	0.34+0%	0.62+37%	0.20+0%	0.34+0%	0.62+38%	0.20+0%	0.34+0%
	LRP	0.89+49%	0.72+115%	0.83+65%	0.89+49%	0.72+115%	0.82+65%	0.89+50%	0.72+115%	0.82+65%
$t_2 = 500$	GradCam	0.88+21%	0.66+61%	0.78+37%	0.88+22%	0.66+61%	0.78+37%	0.88+22%	0.66+61%	0.78+37%
	GBP	0.91+114%	0.28+42%	0.43+33%	0.91+114%	0.28+42%	0.43+33%	0.91+114%	0.28+43%	0.43+33%
	Guide-GC	0.93+17%	0.35+32%	0.51+24%	0.93+17%	0.35+32%	0.51+24%	0.93+18%	0.35+33%	0.51+24%
	IxG	0.62+39%	0.20+0%	0.34+0%	0.62+39%	0.20+0%	0.34+0%	0.63+40%	0.20+0%	0.34+0%
	IG	0.61+36%	0.21+0%	0.34+0%	0.62+37%	0.21+0%	0.34+0%	0.62+38%	0.21+0%	0.34+0%
	LRP	0.90+50%	0.73+117%	0.83+65%	0.90+50%	0.73+117%	0.83+65%	0.90+51%	0.73+117%	0.83+65%

Table 6. Low variation between scaling parameters. Across 11 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

		DenseNet121								
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	GradCam	0.59+17%	0.37+7%	0.49-1%	0.59+17%	0.37+7%	0.49-1%	0.59+17%	0.37+8%	0.49-1%
	GBP	0.83+152%	0.25+27%	0.39+21%	0.83+152%	0.25+26%	0.39+21%	0.83+153%	0.25+26%	0.39+21%
	Guide-GC	0.71+27%	0.26+12%	0.40+9%	0.71+27%	0.26+12%	0.40+9%	0.71+27%	0.26+12%	0.40+9%
	IxG	0.45+28%	0.20+0%	0.33+0%	0.46+28%	0.20+0%	0.33+0%	0.46+29%	0.20+0%	0.33+0%
	IG	0.49+31%	0.20+0%	0.34+0%	0.49+32%	0.20+0%	0.34+0%	0.49+33%	0.20+0%	0.34+0%
	LRP	0.44+23%	0.25+0%	0.40+0%	0.44+24%	0.25+0%	0.40+0%	0.45+25%	0.25+0%	0.40+0%
$t_2 = 100$	GradCam	0.60+17%	0.37+6%	0.48-2%	0.60+17%	0.37+6%	0.48-2%	0.60+17%	0.37+6%	0.49-2%
	GBP	0.84+157%	0.25+27%	0.40+21%	0.85+158%	0.25+27%	0.40+21%	0.85+158%	0.25+27%	0.40+21%
	Guide-GC	0.71+28%	0.26+12%	0.40+9%	0.71+28%	0.26+12%	0.40+9%	0.71+28%	0.26+12%	0.40+9%
	IxG	0.46+31%	0.20+0%	0.33+0%	0.46+31%	0.20+0%	0.33+0%	0.47+32%	0.20+0%	0.33+0%
	IG	0.50+33%	0.20+0%	0.34+0%	0.50+34%	0.20+0%	0.34+0%	0.50+35%	0.20+0%	0.34+0%
	LRP	0.44+24%	0.25+0%	0.40+0%	0.44+24%	0.25+0%	0.40+0%	0.45+25%	0.25+0%	0.40+0%
$t_2 = 500$	GradCam	0.58+14%	0.34+0%	0.45-8%	0.58+14%	0.34+0%	0.46-8%	0.58+14%	0.34+0%	0.46-8%
	GBP	0.87+165%	0.25+30%	0.40+23%	0.87+166%	0.25+30%	0.40+23%	0.87+166%	0.25+30%	0.40+23%
	Guide-GC	0.74+32%	0.26+12%	0.40+9%	0.74+32%	0.26+12%	0.40+9%	0.74+32%	0.26+12%	0.40+9%
	IxG	0.47+32%	0.20+0%	0.33+0%	0.47+33%	0.20+0%	0.33+0%	0.47+33%	0.20+0%	0.33+0%
	IG	0.50+34%	0.20+0%	0.34+0%	0.50+34%	0.20+0%	0.34+0%	0.50+35%	0.20+0%	0.34+0%
	LRP	0.44+23%	0.25+0%	0.40+0%	0.44+24%	0.25+0%	0.40+0%	0.45+25%	0.25+0%	0.40+0%

Table 7. Low variation between scaling parameters. Across 6 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

		ConvNeXT								
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	GradCam	0.96+2%	0.56-6%	0.70-5%	0.96+2%	0.56-6%	0.70-5%	0.96+2%	0.56-6%	0.70-5%
	GBP	0.50+23%	0.20+0%	0.33+0%	0.50+24%	0.20+0%	0.33+0%	0.51+24%	0.20+0%	0.33+0%
	Guide-GC	0.96+1%	0.35+1%	0.52+1%	0.96+1%	0.35+1%	0.52+1%	0.96+1%	0.35+1%	0.52+1%
	IxG	0.50+24%	0.20+0%	0.33+0%	0.50+24%	0.20+0%	0.33+0%	0.50+24%	0.20+0%	0.33+0%
	IG	0.62+32%	0.21+0%	0.34+0%	0.63+33%	0.21+0%	0.34+0%	0.63+33%	0.21+0%	0.34+0%
$t_2 = 100$	GradCam	0.96+2%	0.55-7%	0.70-6%	0.96+2%	0.55-7%	0.70-6%	0.96+2%	0.55-7%	0.70-6%
	GBP	0.51+26%	0.20+0%	0.33+0%	0.52+26%	0.20+0%	0.33+0%	0.52+27%	0.20+0%	0.33+0%
	Guide-GC	0.96+1%	0.35+1%	0.52+1%	0.96+1%	0.35+1%	0.52+1%	0.96+1%	0.35+1%	0.52+1%
	IxG	0.51+26%	0.20+0%	0.33+0%	0.51+27%	0.20+0%	0.33+0%	0.51+27%	0.20+0%	0.33+0%
	IG	0.64+35%	0.21+0%	0.34+0%	0.64+35%	0.21+0%	0.34+0%	0.64+36%	0.21+0%	0.34+0%
$t_2 = 500$	GradCam	0.96+2%	0.55-8%	0.69-7%	0.96+2%	0.55-8%	0.69-7%	0.96+2%	0.55-8%	0.69-7%
	GBP	0.53+31%	0.20+0%	0.33+0%	0.54+31%	0.20+0%	0.33+0%	0.54+32%	0.20+0%	0.33+0%
	Guide-GC	0.97+1%	0.35+1%	0.52+1%	0.97+1%	0.35+1%	0.52+1%	0.97+1%	0.35+1%	0.52+1%
	IxG	0.53+31%	0.20+0%	0.33+0%	0.53+31%	0.20+0%	0.33+0%	0.53+31%	0.20+0%	0.33+0%
	IG	0.64+36%	0.21+0%	0.34+0%	0.65+37%	0.21+0%	0.34+0%	0.65+38%	0.21+0%	0.34+0%

Table 8. Low variation between scaling parameters. Across 5 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

ViT-base-16										
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	Bi-attn	0.94+31%	0.71+180%	0.82+103%	0.94+31%	0.71+180%	0.82+103%	0.94+31%	0.71+180%	0.82+103%
	InFlow	0.86+21%	0.56+126%	0.71+78%	0.86+21%	0.56+126%	0.71+78%	0.86+21%	0.56+126%	0.71+78%
	Grad-Rollout	0.72+73%	0.53+112%	0.68+70%	0.72+73%	0.53+112%	0.68+70%	0.72+73%	0.53+112%	0.68+70%
	T-attn	0.94+32%	0.71+180%	0.82+103%	0.93+32%	0.71+180%	0.82+102%	0.93+32%	0.71+179%	0.82+102%
	LRP	0.77+35%	0.51+105%	0.66+65%	0.77+35%	0.51+105%	0.66+65%	0.77+35%	0.51+105%	0.66+65%
	Gradient	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%
$t_2 = 100$	Bi-attn	0.94+31%	0.71+180%	0.82+103%	0.94+31%	0.71+180%	0.82+103%	0.94+31%	0.71+180%	0.82+103%
	InFlow	0.86+21%	0.56+126%	0.71+78%	0.86+21%	0.56+126%	0.71+78%	0.86+21%	0.56+126%	0.71+78%
	Grad-Rollout	0.73+76%	0.53+113%	0.68+71%	0.73+76%	0.53+113%	0.68+71%	0.73+76%	0.53+113%	0.68+71%
	T-attn	0.94+32%	0.71+180%	0.82+103%	0.93+32%	0.71+180%	0.82+102%	0.93+32%	0.71+179%	0.82+102%
	LRP	0.77+35%	0.51+105%	0.66+65%	0.77+35%	0.51+105%	0.66+65%	0.77+35%	0.51+105%	0.66+65%
	Gradient	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%
$t_2 = 500$	Bi-attn	0.94+31%	0.71+181%	0.82+103%	0.94+32%	0.71+181%	0.82+103%	0.94+31%	0.71+181%	0.82+103%
	InFlow	0.88+23%	0.57+126%	0.71+78%	0.88+23%	0.57+126%	0.71+78%	0.88+23%	0.57+126%	0.71+78%
	Grad-Rollout	0.78+88%	0.55+118%	0.69+73%	0.78+88%	0.54+118%	0.69+73%	0.78+88%	0.54+118%	0.69+73%
	T-attn	0.94+32%	0.71+180%	0.82+103%	0.93+32%	0.71+180%	0.82+102%	0.93+32%	0.71+180%	0.82+102%
	LRP	0.78+36%	0.51+105%	0.66+66%	0.78+36%	0.51+105%	0.66+66%	0.78+36%	0.51+105%	0.66+66%
	Gradient	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%	0.93+4%	0.57+3%	0.70+2%

Table 9. Low variation between scaling parameters. Across 6 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

ViT-base-8										
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	Bi-attn	0.90+47%	0.62+147%	0.75+88%	0.91+47%	0.62+146%	0.75+88%	0.91+48%	0.62+146%	0.75+88%
	InFlow	0.82+18%	0.47+89%	0.63+58%	0.82+18%	0.47+89%	0.63+58%	0.82+18%	0.47+89%	0.63+58%
	Grad-Rollout	0.71+50%	0.45+80%	0.61+53%	0.71+50%	0.45+80%	0.61+53%	0.71+50%	0.45+80%	0.61+53%
	T-attn	0.90+53%	0.63+151%	0.76+90%	0.90+53%	0.63+151%	0.76+90%	0.90+53%	0.63+151%	0.76+90%
	LRP	0.76+25%	0.42+70%	0.58+46%	0.76+25%	0.42+70%	0.58+46%	0.76+25%	0.42+70%	0.58+46%
	Gradient	0.90+7%	0.49+8%	0.64+5%	0.90+7%	0.49+8%	0.64+5%	0.90+7%	0.49+8%	0.64+5%
$t_2 = 100$	Bi-attn	0.91+47%	0.62+149%	0.76+89%	0.91+48%	0.62+149%	0.76+89%	0.91+48%	0.62+149%	0.76+89%
	InFlow	0.82+18%	0.47+89%	0.63+58%	0.82+18%	0.47+89%	0.63+58%	0.82+18%	0.47+89%	0.63+58%
	Grad-Rollout	0.71+51%	0.45+80%	0.61+53%	0.71+51%	0.45+80%	0.61+53%	0.71+51%	0.45+80%	0.61+53%
	T-attn	0.90+54%	0.63+152%	0.76+90%	0.90+53%	0.63+152%	0.76+90%	0.90+53%	0.63+152%	0.76+90%
	LRP	0.76+25%	0.42+69%	0.58+46%	0.76+25%	0.42+69%	0.58+46%	0.76+25%	0.42+69%	0.58+46%
	Gradient	0.90+7%	0.49+7%	0.64+5%	0.90+7%	0.49+7%	0.64+5%	0.90+7%	0.49+7%	0.64+5%
$t_2 = 500$	Bi-attn	0.91+47%	0.63+152%	0.76+90%	0.91+48%	0.63+152%	0.76+90%	0.91+48%	0.63+152%	0.76+90%
	InFlow	0.83+20%	0.47+89%	0.63+58%	0.83+20%	0.47+89%	0.63+58%	0.83+20%	0.47+89%	0.63+58%
	Grad-Rollout	0.74+57%	0.45+81%	0.61+53%	0.74+57%	0.45+81%	0.61+53%	0.74+57%	0.45+81%	0.61+53%
	T-attn	0.91+54%	0.63+152%	0.76+90%	0.90+54%	0.63+152%	0.76+90%	0.90+53%	0.63+152%	0.76+90%
	LRP	0.76+25%	0.42+69%	0.58+46%	0.76+25%	0.42+69%	0.58+46%	0.76+25%	0.42+69%	0.58+46%
	Gradient	0.90+7%	0.49+7%	0.64+4%	0.90+7%	0.49+7%	0.64+5%	0.90+7%	0.49+7%	0.64+5%

Table 10. Low variation between scaling parameters. Across 6 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

		ViT-base-32								
		$t_1 = 2$			$t_1 = 5$			$t_1 = 10$		
	Method	RA	IoU	F1	RA	IoU	F1	RA	IoU	F1
$t_2 = 50$	Bi-attn	0.86+71%	0.62+149%	0.75+87%	0.85+71%	0.62+149%	0.75+87%	0.85+70%	0.62+148%	0.75+87%
	InFlow	0.77+20%	0.56+124%	0.70+75%	0.77+20%	0.56+124%	0.70+75%	0.77+20%	0.56+124%	0.70+75%
	Grad-Rollout	0.64+84%	0.51+103%	0.66+64%	0.64+84%	0.51+102%	0.66+64%	0.64+84%	0.51+102%	0.66+64%
	T-attn	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%
	LRP	0.66+49%	0.46+85%	0.61+53%	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%
	Gradient	0.79+19%	0.51+19%	0.65+12%	0.66+49%	0.46+85%	0.61+53%	0.66+49%	0.46+85%	0.61+53%
$t_2 = 100$	Bi-attn	0.86+71%	0.62+149%	0.75+87%	0.86+71%	0.62+149%	0.75+87%	0.85+71%	0.62+148%	0.75+87%
	InFlow	0.78+21%	0.56+124%	0.70+75%	0.78+21%	0.56+124%	0.70+75%	0.78+21%	0.56+124%	0.70+75%
	Grad-Rollout	0.66+91%	0.51+106%	0.66+66%	0.66+91%	0.51+106%	0.66+66%	0.66+91%	0.51+106%	0.66+66%
	T-attn	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%
	LRP	0.66+49%	0.46+85%	0.61+53%	0.66+49%	0.46+85%	0.61+53%	0.66+49%	0.46+85%	0.61+53%
	Gradient	0.79+19%	0.51+19%	0.65+12%	0.79+19%	0.51+19%	0.65+12%	0.79+19%	0.51+19%	0.65+12%
$t_2 = 500$	Bi-attn	0.86+71%	0.62+149%	0.75+87%	0.86+71%	0.62+149%	0.75+87%	0.85+71%	0.62+149%	0.75+87%
	InFlow	0.80+24%	0.57+126%	0.70+76%	0.80+24%	0.57+126%	0.70+76%	0.80+24%	0.57+126%	0.70+76%
	Grad-Rollout	0.72+110%	0.54+116%	0.68+71%	0.72+110%	0.54+116%	0.68+71%	0.72+110%	0.54+116%	0.68+71%
	T-attn	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%	0.84+70%	0.62+146%	0.74+86%
	LRP	0.68+53%	0.47+86%	0.61+53%	0.68+53%	0.47+86%	0.61+53%	0.68+53%	0.47+86%	0.61+53%
	Gradient	0.79+19%	0.51+18%	0.65+11%	0.79+19%	0.51+18%	0.65+11%	0.79+19%	0.51+19%	0.65+11%

Table 11. Low variation between scaling parameters. Across 6 different attribution methods considering convolutional and transformer based architectures, quantitative metrics measured using Region Attribution (RA), Intersection over Union (IoU), and F1 vary very slightly across various hyperparameter selections.

Table 12. Improving transformer attributions on deletion test. Augmenting the base method with AL improves the AUC (lower is better) for insertion tests for convolutional architectures by 0-14%. GradCAM is again an outlier since it almost deletes the image in one go. For ViTs modestly worsens the AUC, similar to GradCAM attribution methods for ViTs are often very diffuse and large, hence deleting a big part of the image will yield strong results.

(a) CNN-based architectures

Method	ResNet50	WideResNet50-2	DenseNet121	ConvNeXT
IG	0.06–25%	0.06–14%	0.07–22%	0.10–9%
GBP	0.04–20%	0.04–20%	0.04–33%	0.13–13%
IxG	0.09–18%	0.08–20%	0.10–17%	0.13–13%
Guide-GC	0.03–25%	0.04+0%	0.05+0%	0.07+0%
GradCam	0.04+0%	0.05+0%	0.07+0%	0.08+33%
LRP	0.04+0%	0.04+0%	0.11+0%	0.00+0%
Avg improvement	–14.67%	–9.00%	–12.00%	–0.33%

(b) Transformer-based architectures

Method	ViT-base-8	ViT-base-16	ViT-base-32
Bi-attn	0.06+0%	0.05+0%	0.04+0%
T-attn	0.07+0%	0.28+22%	0.05+25%
InFlow	0.06+0%	0.28+22%	0.04+0%
Gradient	0.07+0%	0.06+0%	0.05+25%
Grad-RI	0.07+0%	0.06–14%	0.04–20%
LRP	0.00+0%	0.00+0%	0.00+0%
Avg improvement	0.00%	+5.00%	+5.00%



Figure 7. We show an example of how the combined images look. We can see that both objects are clearly visible and identifiable.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

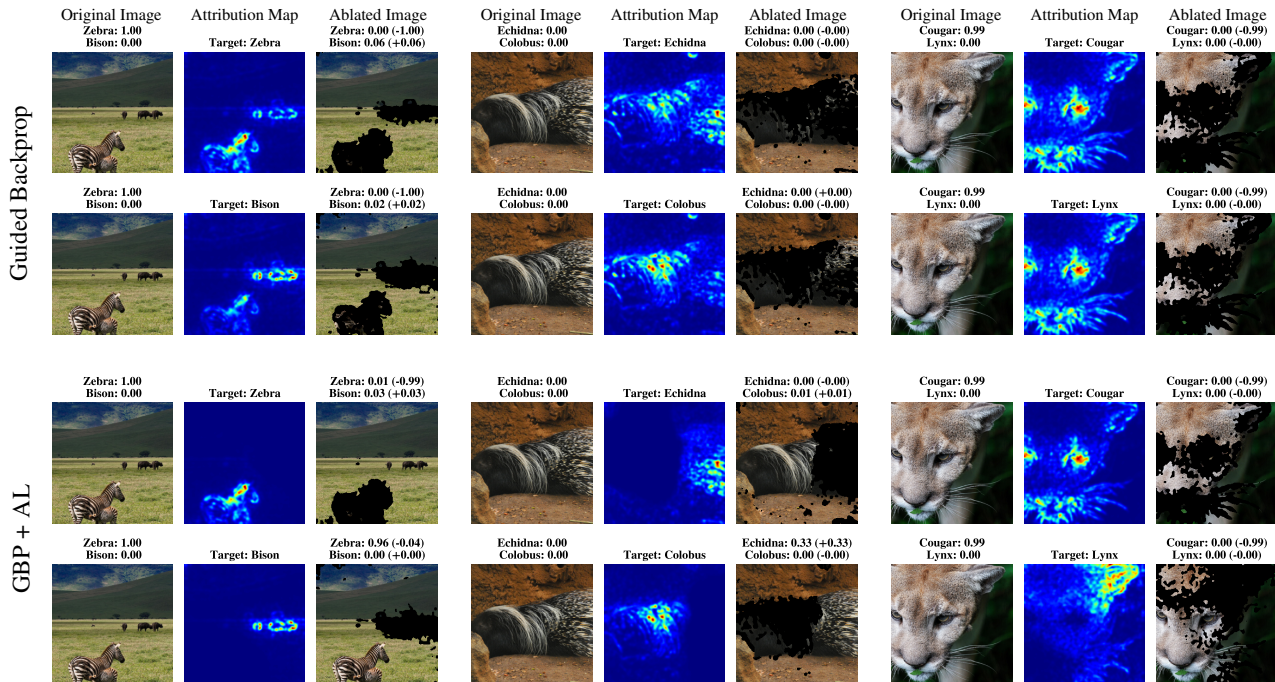


Figure 8. Qualitative example of the ablation study. For GBP (top) and GBP with AL (bottom) we provide examples from the insertion/deletion ablation. For each, we show the original image with class softmax scores for two classes, the attribution map for each of the classes, and the attribution-based intervention mask on each of the classes with resulting changes in class softmax scores.

D.5. Sanity Checks

We show the sanity check plots for these additional architectures in Figure 9-15.

D.6. Ablations

We show more examples of the ablation in Figure 8.

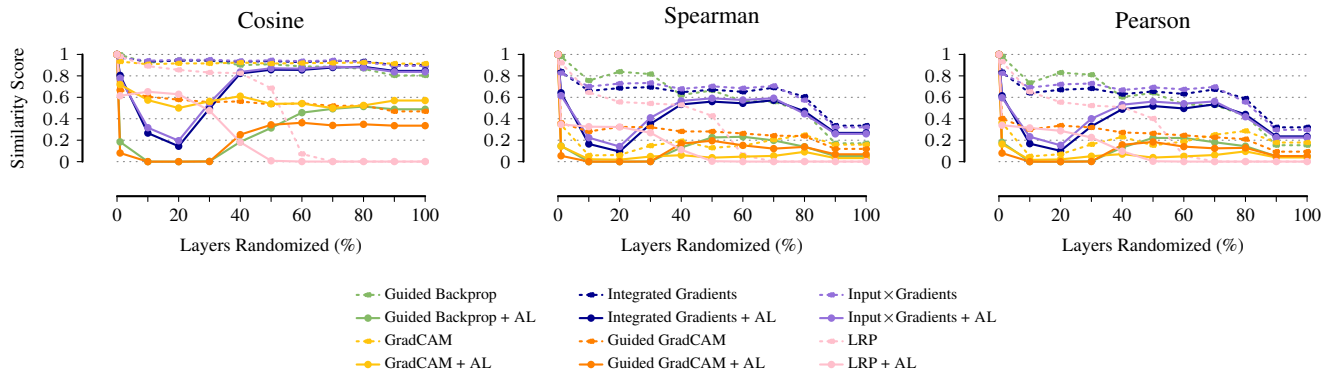


Figure 9. ResNet50: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

Hidden in Plain Sight – Class Competition Focuses Attribution Maps

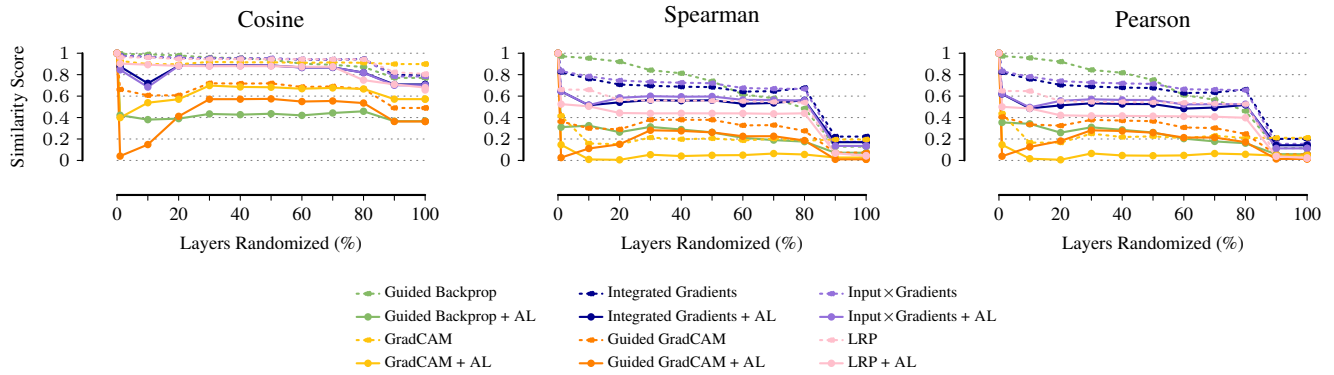


Figure 10. DenseNet121: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

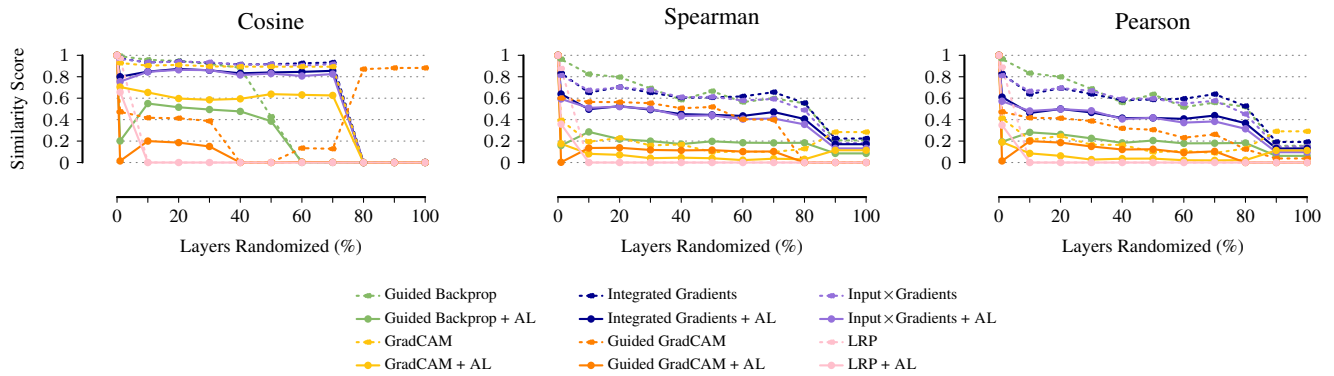


Figure 11. WRN50-2: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

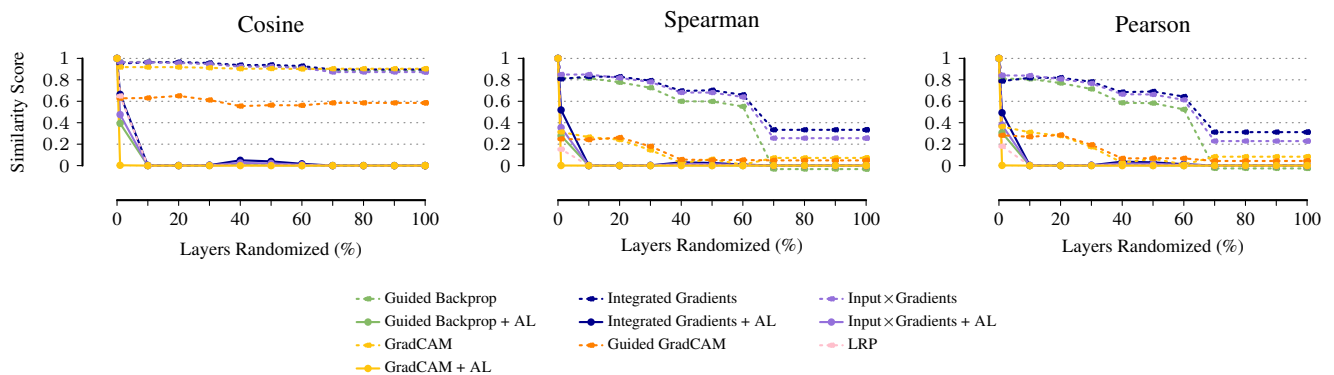


Figure 12. ConvNext: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

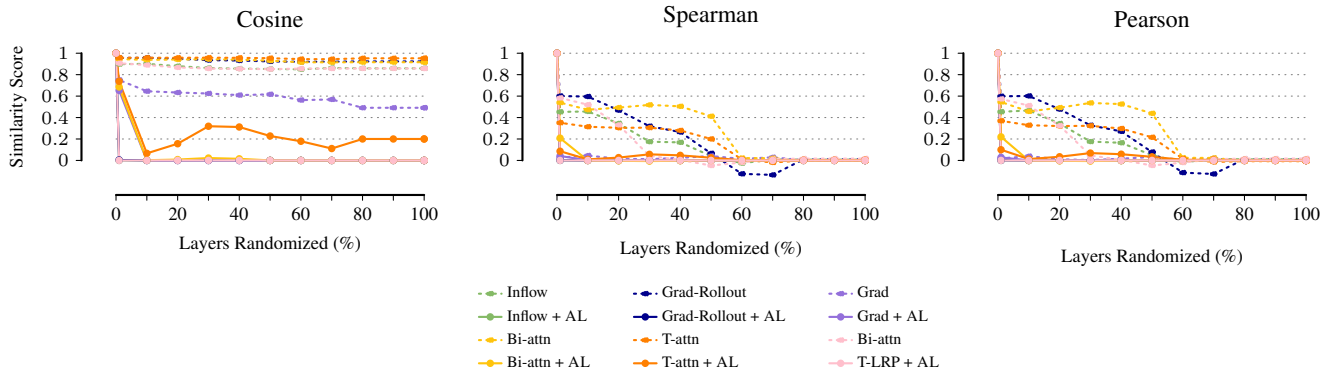


Figure 13. ViT-base-8: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

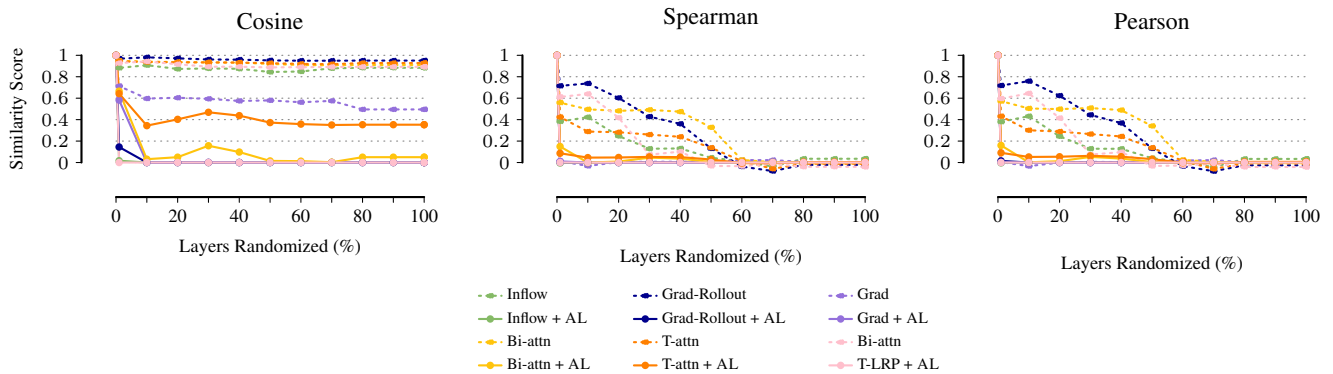


Figure 14. ViT-base-16: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.

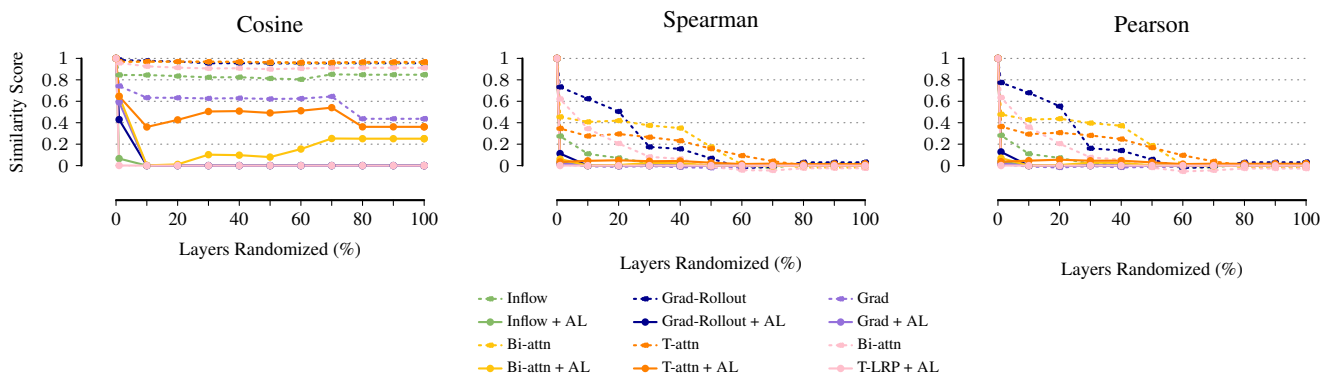


Figure 15. ViT-base-32: AL improves all base methods under randomization [Lower is better]. For all methods and for varying level of randomization, we measure the similarity between the attention map for the unperturbed network and the randomized network. Dashed lines are base methods, solid lines when augmenting with AL, which improve the corresponding baseline method.



Figure 16. **ResNet50**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.



Figure 17. **DenseNet121**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.

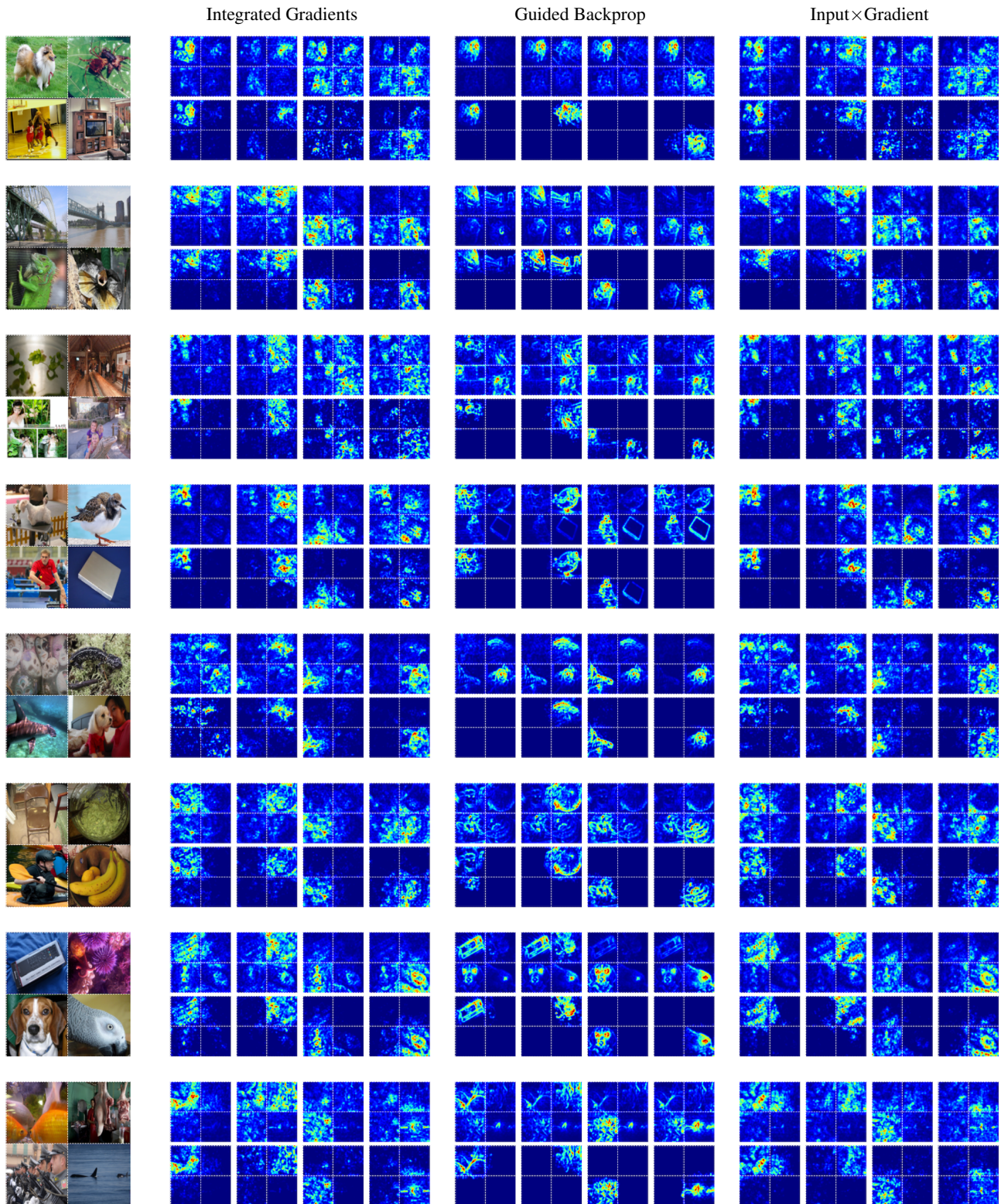


Figure 18. **WideResNet50-2**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.



Figure 19. **ConvNeXt**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input x Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.



Figure 20. **ViT-base-8**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.

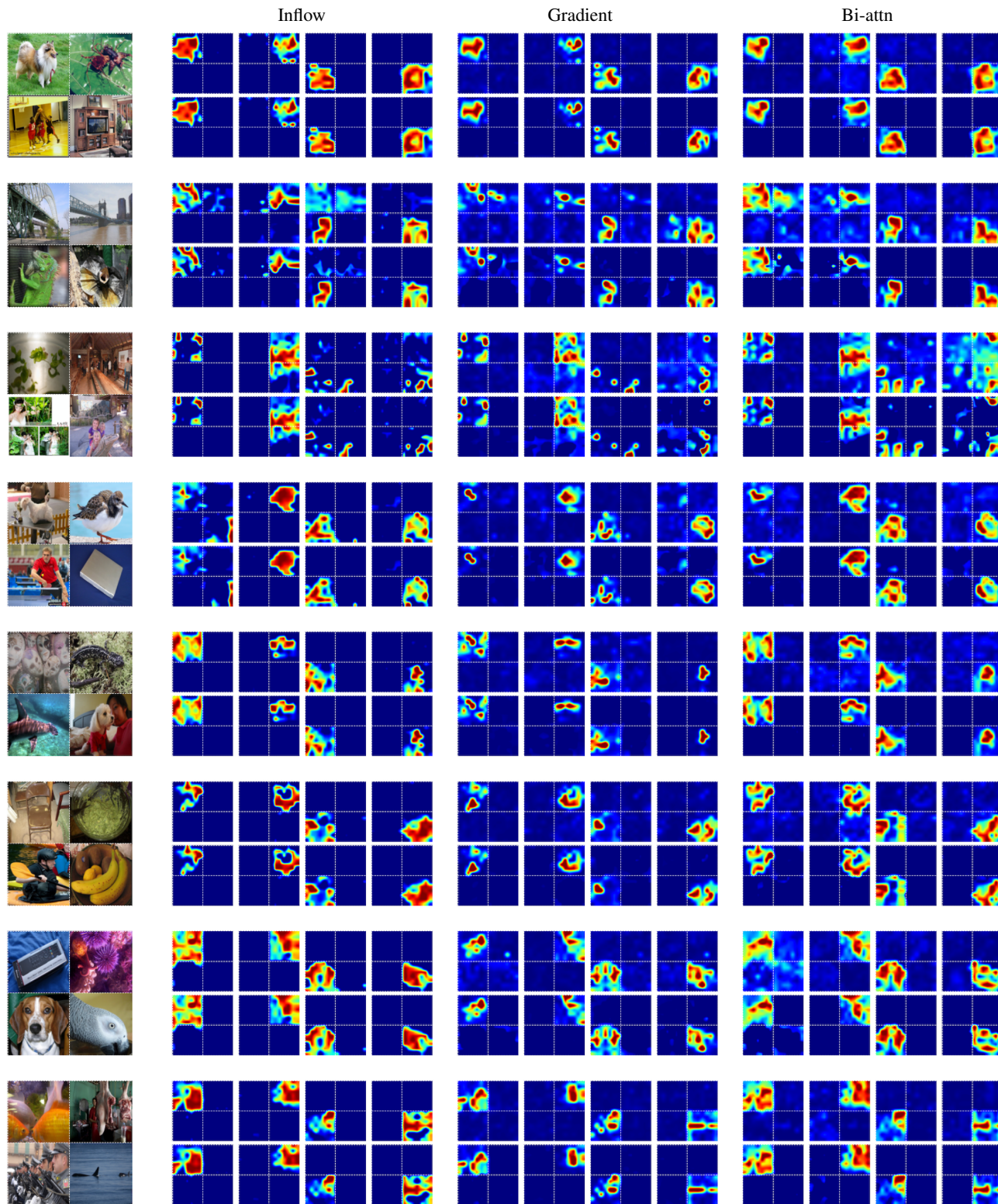


Figure 21. **ViT-base-16**: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.

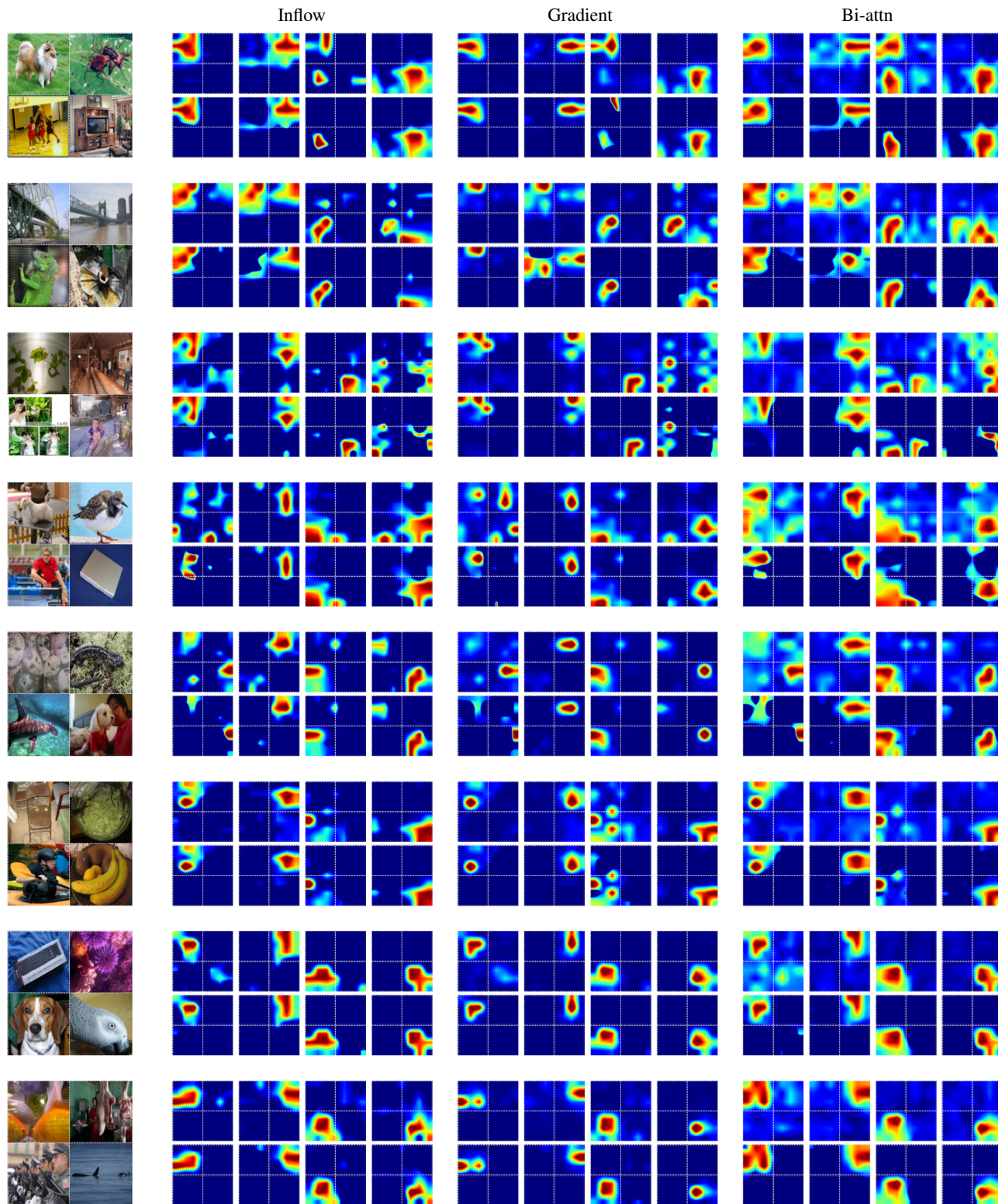


Figure 22. ViT-base-32: AL on the Grid Pointing Game. We show examples from the grid pointing game for methods most affected by our framework (as columns: Integrated Gradient, Guided Backpropagation, Input \times Gradient). Input Images are given on the left, for each we provide vanilla attribution methods (top row) and augmented with AL (bottom row). For each, we show the attribution for the four different classes in the grid as columns.