arXiv:2604.05819v2 [cs.CV] 5 Sep 2026

Learn to Rank: Visual Attribution by Learning Importance Ranking

David Schinagl[1], Christian Fruhwirth-Reisinger[1,2], Alexander Prutsch[1], Samuel Schulter[3⋆], and Horst Possegger[1]

[1] Institute of Visual Computing (IVC), Graz University of Technology, Graz, Austria
{david.schinagl,reisinger,alexander.prutsch,possegger}@tugraz.at
[2] Nomadic AI, USA
[3] Amazon, USA

Abstract. Interpreting the decisions of complex computer vision models is crucial to establish trust and accountability, especially in safety-critical domains. An established approach to interpretability is generating visual attribution maps that highlight regions of the input most relevant to the model's prediction. However, existing methods face a three-way trade-off. Propagation-based approaches are efficient, but they can be biased and architecture-specific. Meanwhile, perturbation-based methods are causally grounded, yet they are expensive and for vision transformers often yield coarse, patch-level explanations. Learning-based explainers are fast but usually optimize surrogate objectives or distill from heuristic teachers. We propose a learning scheme that instead optimizes deletion and insertion metrics directly. Since these metrics depend on non-differentiable sorting and ranking, we frame them as permutation learning and replace the hard sorting with a differentiable relaxation using Gumbel-Sinkhorn. This enables end-to-end training through attribution-guided perturbations of the target model. During inference, our method produces dense, pixel-level attributions in a single forward pass with optional, few-step gradient refinement. Our experiments demonstrate consistent quantitative improvements and sharper, boundary-aligned explanations, particularly for transformer-based vision models. Code and pre-trained models are available at `https://github.com/dschinagl/AHA`.

Keywords: Visual Attribution · Amortized Explanation · Permutation

1 Introduction

Over the past decade, deep neural networks [25,33,41,42] have powered the vast majority of vision models. The growing complexity of these inherently opaque models, however, makes interpreting their decisions difficult. Especially in safety-critical domains such as healthcare or autonomous driving, explaining *why* a model predicts a certain outcome can be as crucial as the accuracy of the prediction itself. A well-established approach for explainability in computer vision

⋆ This work is independent of the author's employment at Amazon.

is the creation of visual attribution maps, which highlight the input regions most relevant to a model's prediction [14,48,55,58,72]. Such attribution maps provide an intuitive, human-interpretable way to understand model behavior and can help identify potential biases or failure modes [4, 11].

The majority of attribution methods are *post-hoc*, *i.e.* they explain already trained models without modifying their architecture. These can be broadly categorized into three groups: *Propagation-based* approaches which rely on propagating internal signals through the target model, such as gradients [58, 64, 72], activations [55, 68, 74] or attention weights [1, 14, 15] in transformer-based vision models. They are very efficient and align with the model's internal computations but are architecture-specific, can bias toward low-level features like edges [2, 61] and may under-represent non-attention pathways in transformer models. In contrast, *perturbation-based* methods [30, 43, 48, 52] infer feature relevance from changes in the target model's output when parts of the input are modified. They provide more causally grounded attributions, as they directly measure the effects of input changes, including higher-order interactions. However, they are computationally demanding, requiring many model evaluations to produce a single attribution map. For transformer-based vision models, they usually operate on the native patch representation [27, 66, 69] yielding patch-level explanations that must be upsampled to pixel space, resulting in coarse, spatially imprecise maps, as shown in Figure 1. The third and least explored category, *learning-based* (amortized) explainers shift the computational burden to an offline training phase by learning a separate explainer model, which is then able to generate attribution maps very efficiently during inference [21]. However, the current training strategies typically rely on either supervision from other attribution methods [34, 60] and may adopt their biases, or on self-supervised surrogate objectives that are based on restrictive assumptions like a binary signal to noise decomposition of features [7, 18].

Motivated by these limitations, it is natural to ask whether we can avoid attribution heuristics and instead *optimize attribution quality directly*. Common and widely-used attribution metrics are the *Deletion* and *Insertion* metrics [48]. They quantify the change in a model's output when features are removed or added back in order of importance. Recent metric-driven approaches [20, 66] for vision transformers pursue this direction, but do so via a perturbation-based, per-sample optimization on the patch representation. Consequently, they retain the high computational cost and cannot refine the coarse spatial resolution.

In this work, we propose *Amortized Hybrid Attribution (AHA)*: a learning-based approach that bridges these families of attribution methods by integrating their complementary strengths into a unified framework. We retain the causal grounding of perturbation-based methods by learning from the target model's outputs under attribution-guided perturbations. Since we couple explainer and target model through end-to-end backpropagation, we inherit the model-consistent attribution characteristic of propagation-based approaches. We realize this by directly optimizing the *Deletion* and *Insertion* metrics. To overcome the non-differentiability of these metrics, caused by the sorting operations

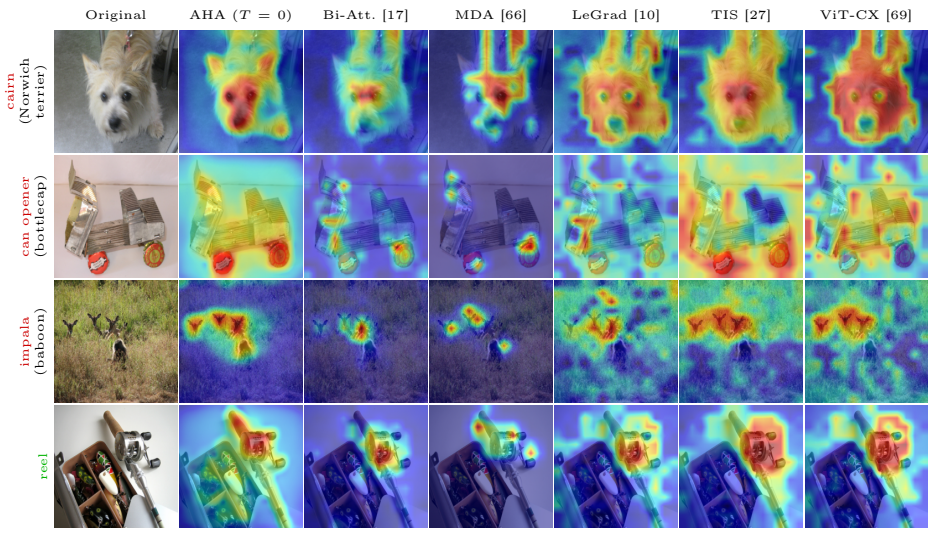


Fig. 1: Qualitative comparison of attribution maps produced by our method (AHA) and recent ViT attribution methods for a ViT-B/16 classifier [25] on ImageNet [22] samples. The green / red labels mark correct / incorrect predictions (ground truth in parentheses). While existing approaches operate on the patch representation and produce coarse, blocky attribution maps, our method generates dense, pixel-level attributions that align closely with object boundaries. Note that in this case no additional backprop refinement steps were applied at inference (denoted $T = 0$).

they depend on, we leverage the observation that they depend solely on the order of pixel importance rather than their exact values, and treat them as permutation problems. By replacing the hard sorting and ranking operations with smooth approximations, we enable end-to-end training of the explainer model. At inference, we preserve the amortized efficiency of learned explainers (single forward pass), but we also keep the door open to the propagation-based regime: we can optionally perform a few additional backward steps through the target model to locally refine the explainer's output on a specific sample. This effectively combines a learned, metric-aligned prior with a cheap, model-specific propagation signal. Due to the end-to-end optimization our explainer is able to generate dense attributions that align closely with object boundaries and visually meaningful structures, even for transformer-based vision models, as shown in Figure 1.

Our contributions can be summarized as follows:

- We propose a learning-based attribution framework that bridges perturbation- and propagation-based explainability.
- Our amortized explainer directly optimizes differentiable surrogates of *Deletion* and *Insertion*, producing dense pixel-space attributions also for transformer-based vision models.

- We formulate metric optimization as a permutation learning problem and enable end-to-end training via a Gumbel-Sinkhorn relaxation of sorting and top-k selection.
- To reduce the computational burden and mitigate pixel-level shortcut artifacts, the training procedure leverages region-level permutation, perturbation-step sampling, and grid augmentation.
- At inference, the method retains single-pass efficiency and optionally adds a few backprop refinement steps for sample-specific adjustments.

2 Related Work

Visual attribution methods highlight the most influential input regions for a given model decision. *Ante-hoc* methods [3] alter the model architecture or training, *e.g.*, prototype-based networks [16, 24, 40, 46], concept bottleneck models [39, 47], or alignment-based approaches [5, 9, 65]. The vast majority, however, are *post-hoc* methods that explain already trained models without modifying them, typically via *propagation-*, *perturbation-* or *learning-based* methods.

Propagation-based Methods propagate attribution signals through the model to determine the relevance of input features. *Backpropagation*-based methods use gradients of the output w.r.t. the input [58, 72]. Various enhancements mitigate noisy gradients, including aggregating multiple gradient signals [37,38,63,64,70], preventing negative gradients from flowing back [62], or specific backpropagation rules [6,56,73]. *Forward propagation*-based methods, on the other hand, utilize the activations of intermediate layers to generate attribution maps [36, 74]. Grad-CAM [55] and its successors [13, 35] use the gradient w.r.t. the feature maps to weight the activations in the last convolutional layers. Specialized approaches to interpret vision transformers [12, 25, 32, 41, 50] typically analyze the self-attention mechanism: Attention-Rollout [1] recursively multiplies attention weights across layers, Attention-Flow [1] models attention as a flow network, and Transition-Attention [71] accumulates attention as a Markov process combined with Integrated Gradients [64]. Based on LRP [6], Transformer-Attribution [15] proposes propagation rules with gradient integration for self-attention layers. To approximate token contributions, Bidirectional Attention [17] jointly analyzes attention and partial derivatives to produce interpretable attribution signals. Instead of using the gradients to weight the attention maps, LeGrad [10] uses the gradient w.r.t. the attention scores as attribution signals.

The major advantages of these approaches are computational efficiency and their model faithfulness as the attribution is directly derived from the model's internal computations. However, they are often tied to specific architectures or layer types and the reliance on local gradients can bias the attributions toward low-level edges and textures [2, 61].

Perturbation-based Methods directly measure output changes under systematic input perturbations to create causally grounded attributions [75]. In their pioneering work, Zeiler and Fergus [72] monitored the model's output

change while a sliding window occludes parts of the input. LIME [52] uses superpixel perturbations to fit an interpretable surrogate model per sample, whereas SHAP [43] leverages cooperative game theory to assign attributions by approximating Shapley values over feature subsets for individual predictions. Fong *et al*. [29,30] formulate attribution as a per-sample optimization problem, searching for the smallest region that maximally influences the output. RISE [48] generates attribution maps as an output score-weighted combination of random input masks, while EVA [28] proposes a more systematic and exhaustive exploration of the perturbation space. While these perturbation-based approaches have predominantly been studied for CNNs, recent adaptations exploit the patch-wise input representation of vision transformers: ViT-CX [69] derives attribution by scoring the causal impact of patch tokens on the model output, while TIS [27] leverages the variable-length token processing of transformers by removing token subsets. Contrary to heuristic attribution generation methods, Metric-Driven Attributions (MDA) [66] aims to directly optimize the metrics used to evaluate the attribution maps. In a per-sample optimization process, MDA aims to find the optimal order and magnitude of patch perturbations to maximize the evaluation metrics. Relatedly, *Less is More* [20] builds on a given attribution map as a prior and refines it via greedy submodular selection of few regions, resulting in a small set of explanatory masks.

Since perturbation-based methods directly measure the model responses, they provide causally grounded attributions and can capture higher-order feature interactions. However, a major drawback is that they are computationally expensive as multiple model evaluations are required per attribution map. Moreover, methods tailored to vision transformers typically yield only patch-level attributions and therefore rely on upsampling to obtain input-resolution maps, often resulting in coarse and spatially imprecise attribution maps (recall Fig. 1).

Learning-based Methods (Amortized Explainers) train a neural explainer model to directly output attributions [21]. This amortizes the cost of per-sample iterative computation by shifting it to an offline training stage performed once per target model, while explanations can be generated efficiently at inference time. *Prior explanation*-driven approaches use other attribution methods as supervisory signals [60] or approximate Shapley-based priors [34]. *Self-supervised amortizers* on the other hand, interact with the target model to optimize a proxy objective. However, these objectives often rely on ideal assumptions like a near-binary signal/noise decomposition to isolate relevant features [7], or assume that only a small, separable feature subset carries the predictive signal [18], which may not hold in practice, especially for complex vision models.

In contrast, our learning-based approach is directly guided by recognized evaluation metrics (*Deletion* & *Insertion*) without any heuristic assumptions about the underlying data distribution or model behavior. Since the supervisory signal is derived from the model's responses to perturbations, our method remains causally grounded. Additionally, our end-to-end optimization schema is not limited to patch-level attribution, resulting in denser and better aligned results than existing ViT-specific methods.

3 Amortized Hybrid Attribution

In the following, we first formalize the problem setting to obtain attribution maps produced for a fixed, pretrained classifier. We then review the non-differentiable perturbation-based *Deletion* and *Insertion* metrics that serve as our target notion of attribution quality, and present our differentiable relaxation. This makes the involved sorting and top-k selection operations differentiable, allowing us to optimize the metrics end-to-end. Finally, we describe practical strategies to make training efficient and robust at high resolution, as well as an optional test-time refinement to adapt the attribution map to sample-specific characteristics for use cases which prioritize explanation quality over throughput.

Problem Setting and Notation. Let $f : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^C$ denote a pretrained image classifier that maps an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ to class probabilities $f(\mathbf{I}) = \mathbf{s} \in \mathbb{R}^C$, where s_c corresponds to the predicted probability of class c. Given a target class $t \in \{1, \dots, C\}$, an attribution method is defined as a mapping $\Phi_f : \mathbb{R}^{H\times W\times 3} \times \{1, \dots, C\} \to \mathbb{R}^{H\times W}$, which produces an attribution map $\mathbf{A} = \Phi_f(\mathbf{I}, t)$ of the same spatial resolution as the input, where each element a_{ij} reflects the importance of spatial location (i, j) to the target probability s_t. Unless otherwise specified $t = \arg\max_c s_c$.

Metrics. Attribution methods are commonly evaluated by perturbation metrics measuring changes in model output under attribution-guided input perturbations. Petsiuk *et al.* [48] introduced the *Deletion* and *Insertion* metrics, which are widely used for attribution assessment [10, 14, 15, 19] and optimization [66]. *Deletion* quantifies the drop in the target probability as high-attribution pixels are removed, while *Insertion* measures the probability increase as these pixels are added to a blank reference image. Formally this is defined as follows: Let $\mathbf{a} = \mathrm{vec}(\mathbf{A}) \in \mathbb{R}^N$ with $N = H \cdot W$ denote the vectorized attribution map. First, a permutation $\pi_{\mathbf{A}}$ over the pixel indices is computed such that it sorts the entries of $\mathbf{a}$ in descending order, $a_{\pi_{\mathbf{A}}(1)} \geq a_{\pi_{\mathbf{A}}(2)} \geq \cdots \geq a_{\pi_{\mathbf{A}}(N)}$. Based on this order a sequence of binary masks $\{\mathbf{M}_{\mathbf{A}}^{(k)}\}_{k=1}^{N}$ is constructed, where each mask $\mathbf{M}_{\mathbf{A}}^{(k)} \in \{0, 1\}^{H\times W}$ selects the top-k pixels. Next, the input image $\mathbf{I}$ is progressively blended with a blank reference image $\mathbf{I}_0$ (e.g., black image, mean colored image or a blurred version of the original image) using these masks, yielding two sequences of perturbed images $\{\mathbf{I}_{del}^{(k)}\}_{k=1}^{N}$ and $\{\mathbf{I}_{ins}^{(k)}\}_{k=1}^{N}$ where,

$$\mathbf{I}_{del}^{(k)}(\mathbf{A}, \mathbf{I}) = \mathbf{I} \odot (1 - \mathbf{M}_{\mathbf{A}}^{(k)}) + \mathbf{I}_0 \odot \mathbf{M}_{\mathbf{A}}^{(k)}, \tag{1}$$

$$\mathbf{I}_{ins}^{(k)}(\mathbf{A}, \mathbf{I}) = \mathbf{I} \odot \mathbf{M}_{\mathbf{A}}^{(k)} + \mathbf{I}_0 \odot (1 - \mathbf{M}_{\mathbf{A}}^{(k)}), \tag{2}$$

with $\odot$ denoting element-wise multiplication. In $\mathbf{I}_{del}^{(k)}$, the top-k pixels of the original image are replaced by the blank reference image, while in $\mathbf{I}_{ins}^{(k)}$, the top-k pixels are inserted into the blank reference image. The attribution quality is then evaluated by computing the area under the curve (AUC) of the target class

probability s_t across the sequence of perturbed images:

$$del_{\text{AUC}}(\mathbf{A}, \mathbf{I}, t) = \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{I}_{del}^{(k)}(\mathbf{A}, \mathbf{I}))_t, \tag{3}$$

$$ins_{\text{AUC}}(\mathbf{A}, \mathbf{I}, t) = \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{I}_{ins}^{(k)}(\mathbf{A}, \mathbf{I}))_t. \tag{4}$$

Meaningful attribution maps lead to a rapid score drop under *Deletion* (low del_{AUC}) and a rapid score increase under *Insertion* (high ins_{AUC}), indicating that highly ranked pixels indeed have strong influence on the prediction.

3.1 Learn to Attribute

In our method we aim to train an attribution model $\Phi_{f,\theta}$ with parameters θ for a given model f, that is able to create dense attribution maps $\mathbf{A}$ which follow the definition of importance as measured by the *Deletion* and *Insertion* metrics. However, the metrics defined in Equations (3) and (4) are non-differentiable w.r.t. the attribution map $\mathbf{A}$ due to the discrete pixel sorting (permutation) $\pi_{\mathbf{A}}$ and the corresponding binary top-k masks $\mathbf{M}_{\mathbf{A}}^{(k)}$, which introduce discontinuities whenever the order changes. Nevertheless, a key observation is that both metrics depend exclusively on the *relative order* of the attribution values, not their absolute values. This fact allows us to bypass the non-differentiability by employing continuous relaxations of the sorting and selection operations, using differentiable sorting techniques [8, 44, 49] to obtain smooth approximations. This relaxation yields soft permutation matrices and, in turn, soft top-k masks that enable end-to-end optimization of del_{AUC} and ins_{AUC} with respect to the attributions.

Soft Permutation Matrices. The hard permutation $\pi_{\mathbf{A}}$ induced by sorting the attribution scores $\mathbf{a}$ can be represented as permutation matrix $\mathbf{P}_{\pi_{\mathbf{A}}} \in \{0,1\}^{N \times N}$ such that,

$$\mathbf{P}_{\pi_{\mathbf{A}}} \mathbf{a} = \left[a_{\pi_{\mathbf{A}}(1)},\, a_{\pi_{\mathbf{A}}(2)},\, \ldots,\, a_{\pi_{\mathbf{A}}(N)}\right]^{\top}. \tag{5}$$

Our approach to obtain a differentiable approximation of this permutation matrix is based on the *Gumbel-Sinkhorn* algorithm [44]. First, we compute a similarity matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$, which measures the similarity between the attribution scores $\mathbf{a}$ produced by the explainer model and a set of N target positions $\mathbf{p} = \frac{1}{N}[N, N-1, \ldots, 1]^{\top}$ defined as linearly decreasing values, representing the ideal sorted distribution. Each entry $\mathbf{L}_{i,j}$ is computed as the negative squared distance between the attribution score a_i and the target position p_j,

$$\mathbf{L}_{i,j} = -(a_i - p_j)^2, \tag{6}$$

where larger values correspond to smaller assignment costs and therefore better matches between attribution scores and target ranks. Subsequently, we add noise

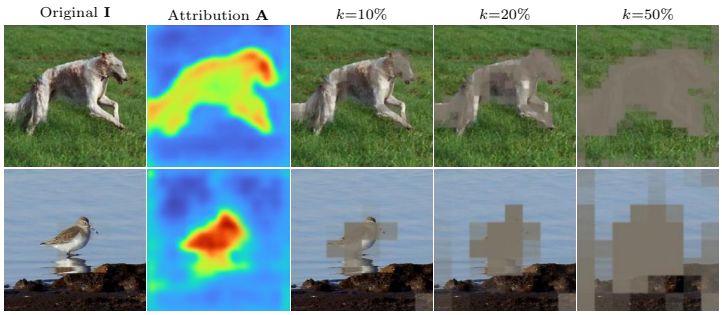


Fig. 2: Soft deletion masks. Given an input image $\mathbf{I}$ and its attribution map $\mathbf{A}$, the soft top-k masks $\mathbf{m}_{\mathbf{A}}^{\text{soft},(k)}$ progressively replace the most important regions with the reference image. Unlike hard binary masks, the soft relaxation (Sec. 3.1) assigns continuous importance values to each region, enabling gradient-based optimization of the deletion and insertion metrics. Note the varying grid resolution G across examples top *vs.* bottom (Sec. 3.2) which prevents overfitting to a fixed spatial partitioning.

sampled from the Gumbel distribution $\mathbf{G} \sim \text{Gumbel}(0, 1)$ to introduce stochasticity, encouraging the exploration of different permutations during training,

$$\widetilde{\mathbf{L}} = \frac{\mathbf{L} + \mathbf{G}}{\tau}, \tag{7}$$

where $\tau > 0$ is a temperature parameter that controls the smoothness of the resulting approximation. This formulation frames the sorting task as an *optimal transport (OT)* problem, seeking the minimum-cost assignment between the attribution scores and the target positions. By applying the *Sinkhorn-Knopp* algorithm [59] to the matrix $\widetilde{\mathbf{L}}$, which performs iterative row and column normalization, we obtain a doubly-stochastic matrix $\mathbf{P}_{\pi_{\mathbf{A}}}^{\text{soft}} \in [0, 1]^{N \times N}$ that serves as a continuous relaxation of the permutation matrix $\mathbf{P}_{\pi_{\mathbf{A}}}$.

Soft Top-k Masks. A hard top-k mask $\mathbf{M}_{\mathbf{A}}^{(k)} \in \{0, 1\}^{H \times W}$ selects all pixels whose rank is between 1 and k. The vectorized form of this mask $\mathbf{m}_{\mathbf{A}}^{(k)} \in \{0, 1\}^{N}$ is created by summing the first k rows of the permutation matrix $\mathbf{P}_{\pi_{\mathbf{A}}}$. Similarly, given our soft permutation matrix $\mathbf{P}_{\pi_{\mathbf{A}}}^{\text{soft}}$, we create a sequence of N soft top-k masks $\{\mathbf{m}_{\mathbf{A}}^{\text{soft},(k)}\}_{k=1}^{N}$, where $\mathbf{m}_{\mathbf{A}}^{\text{soft},(k)} \in [0, 1]^{N}$ is obtained by computing the cumulative sum along the rank dimensions (rows) of the soft permutation matrix

$$\mathbf{m}_{\mathbf{A}}^{\text{soft},(k)} = \sum_{i=1}^{k} \mathbf{P}_{\pi_{\mathbf{A}}}^{\text{soft}}[i, :]^{\top}. \tag{8}$$

This results in masks with continuous values between 0 and 1, indicating the degree to which each pixel is included in the top-k selection, illustrated in Fig. 2.

Differentiable *Deletion & Insertion*. Given the sequence of soft top-k masks, we now replace the hard masks $\mathbf{M}_{\mathbf{A}}^{(k)}$ in Equations (1) and (2) with the

reshaped soft masks $\mathbf{m}_{\mathbf{A}}^{\text{soft},(k)}$ to obtain $\mathbf{I}_{del}^{\text{soft},(k)}$ and $\mathbf{I}_{ins}^{\text{soft},(k)}$. Consequently, we get differentiable approximations of the *Deletion* and *Insertion* metrics as:

$$del_{\text{AUC}}^{\text{soft}}(\mathbf{A},\, \mathbf{I},\, t) = \frac{1}{N}\sum_{k=1}^{N} f\left(\mathbf{I}_{del}^{\text{soft},(k)}(\mathbf{A},\mathbf{I})\right)_t, \tag{9}$$

$$ins_{\text{AUC}}^{\text{soft}}(\mathbf{A},\, \mathbf{I},\, t) = \frac{1}{N}\sum_{k=1}^{N} f\left(\mathbf{I}_{ins}^{\text{soft},(k)}(\mathbf{A},\mathbf{I})\right)_t. \tag{10}$$

The loss function for training the attribution model can then be defined as a weighted combination of these two metrics, encouraging the model to produce attribution maps guided directly by attribution quality measures:

$$\mathcal{L}(\theta) = \lambda_{\text{del}} \cdot del_{\text{AUC}}^{\text{soft}}(\mathbf{A},\, \mathbf{I},\, t) - \lambda_{\text{ins}} \cdot ins_{\text{AUC}}^{\text{soft}}(\mathbf{A},\, \mathbf{I},\, t), \tag{11}$$

where $\lambda_{\text{del}}, \lambda_{\text{ins}}$ are weighting hyperparameters.

3.2 Efficient and Robust Optimization

Although the differentiable relaxations allow us to compute gradients with respect to the attribution maps, a naive implementation of the training process that involves computing the full sequence of N perturbations for each image would be computationally prohibitive, especially for high-resolution images. On the other hand, optimizing the loss described in Equation (11) on pixel-level would be prone to overfitting, as the attribution model could exploit high-frequency artifacts, resulting in directions targeting the architecture-dependent weaknesses of the model under examination. To mitigate these issues, we propose strategies making the training process practical and robust:

- **Region-Based Permutation.** Instead of sorting all N pixels individually, we partition the image into $K \ll N$ disjoint regions using a regular grid of size $K = G \cdot G$ and mean-pool the attribution values within each region. After the differentiable sorting and soft top-k selection on these region-level scores, the resulting masks are upsampled to the original image resolution for perturbation. This significantly reduces the number of perturbations during training and reduces the risk of exploiting individual pixel artifacts.
- **Sampling Perturbation Steps.** To further reduce the computational burden during training, we do not evaluate the full sequence of K perturbation steps for each image. Instead, we uniformly sample a smaller set of $S \ll K$ steps from the range $\{1, \ldots, K\}$, *i.e.*, we perturb multiple grid regions at once instead of adding them one by one. This approximation of the AUC metrics decreases the number of needed forward passes through the model to a fixed budget.
- **Grid Augmentation.** While the grid-based region partitioning enhances the training efficiency and robustness, a fixed grid alignment may introduce bias towards specific spatial patterns. To alleviate this effect we randomly

vary the grid size G within a predefined range for each training iteration and apply random offsets to the grid's position. These augmentations encourage the attribution model to focus on generalizable importance patterns that are not tied to specific grid configurations or scales.
- **Regularization.** In addition to the objective in Equation (11), we regularize the attribution model to discourage spurious, isolated peaks, *i.e.* individual pixels or small clusters of pixels with high scores that are not supported by their neighboring regions. Concretely, we convolve the attribution map $\mathbf{A}$ with an averaging filter and penalize large deviations from the original map using an L_2 loss.

Taken together, these strategies do not merely make training tractable, they also shape which attribution patterns the objective rewards. A natural concern is that the order-based metric could be satisfied by diffuse maps, since only the relative ranking matters in Equations (3) and (4). The optimization, however, does not act on the ordering alone. The Sinkhorn assignment matches the continuous attribution values to the target ranks through the distance costs in Equation (6), so the degree of separation between region scores controls the sharpness of the resulting soft top-k masks. Near-tied scores thus yield ambiguous masks that mix foreground and background, whereas clearly separated scores yield decisive masks. This separation is what the AUC objectives reward: *Deletion* and *Insertion* are driven by a steep score change already at small k, which only arises when the perturbation is concentrated on the few truly causal regions. Decisive masks therefore induce low *Deletion* and high *Insertion*, while diffuse masks perturb the whole image weakly and flatten both curves. Together with the region-level perturbations, the randomized grid sizes and offsets, and the regularizer penalizing isolated peaks, this drives the explainer toward compact, object-aligned attributions.

3.3 Optional Test-Time Refinement

The design of our attribution framework allows for an optional, fast test-time refinement step at low computational cost, that can further enhance the quality of the generated heatmaps on a per-image basis. The key idea is to briefly fine-tune the explainer for a few gradient steps on the specific test image using the same differentiable AUC objectives used during training. To make refinement more stable and deterministic, we use soft sorting without Gumbel noise and fix the sorting temperature to the final training value, *i.e.* $\tau = \tau_{\text{final}}$. After T optimization steps, a final forward pass produces the refined attribution map. This refinement is particularly useful when explanation quality is prioritized over throughput, with the compute-quality trade-off being adjustable via the number of refinement steps. Moreover, if inputs are expected to significantly deviate from the training distribution, this test-time refinement can help to adapt the explainer to the specific characteristics of the new input, improving attribution quality in such cases.

4 Experiments

4.1 Implementation Details

Unless stated otherwise, we use the following setup for all experiments throughout this section. We train our attribution model on the ImageNet-1K training split [22] to explain a frozen ImageNet ViT-B/16 classifier [25]. Our explainer is an encoder-decoder model with a frozen DINOv3 [57] ViT-L/16 backbone and a DPT-inspired decoder [51] conditioned on a learned class embedding. We optimize only the explainer head parameters and keep the DINOv3 backbone frozen. To create perturbation masks during training, we average-pool the heatmap to a $G\times G$ grid and rank regions by mean attribution using Gumbel-Sinkhorn soft sorting with 30 Sinkhorn iterations. During training, G is sampled uniformly from $\{7, \ldots, 28\}$ to reduce grid-alignment bias (region sizes 32–8 px for 224×224 inputs). The *Deletion* and *Insertion* curves are approximated using $S = 16$ uniformly spaced perturbation steps during training. We equally weight the corresponding losses $\lambda_{\text{del}} = \lambda_{\text{ins}} = 1$ and use $\lambda_{\text{reg}} = 2.5\times10^{-3}$ for the pixel regularization term to penalize isolated high-amplitude activations. The explainer is trained for one epoch with AdamW (lr 3×10^{-4}, weight decay 10^{-3}), gradient clipping (norm 1.0), and a one-cycle cosine schedule. In the optional refinement variant, we update only the explainer head for T steps (lr 10^{-4}) on the test image using the same loss, fixed $\tau{=}\tau_{\text{final}}$, and deterministic soft sorting (Gumbel noise disabled).

For evaluation we use the full corresponding validation split (50K images, 1,000 classes) without Test-Time Refinement ($T = 0$), unless explicitly denoted. We explain the *top-1 predicted class* regardless of the output probability, as well as the *ground truth class* if different, and report results separately for both targets. All metrics are averaged over three different types of reference images $\mathbf{I}_0$ used for the perturbation-based evaluations to replace masked-out pixels: a black image, a uniform gray image (ImageNet mean), and a blurred version of the input image. The detailed evaluation results for all metrics, individual reference images, and architectures are reported in the supplementary material.

4.2 Faithfulness Evaluation

Besides the *Deletion* and *Insertion* described in Section 3, we evaluate the faithfulness of the generated attribution maps also using the *Positive* & *Negative* perturbation evaluation [15]: across the evaluation set, the most-attributed regions (*Positive*) or the least-attributed regions (*Negative*) are progressively removed and the target model's top-1 accuracy is tracked as a function of the percentage of perturbed pixels. Low *Positive* AUC values and high *Negative* AUC values indicate that highly ranked regions capture important features for the target model, while low-ranked regions are non-informative. We also report the *Average Drop Percentage* (ADP) and the *Percentage of Increase in Confidence* (PIC) [13]: for each sample an element-wise product between the normalized attribution map ($\in [0, 1]$) and the input image is created to mask the background regions based

Table 1: Quantitative comparison of attribution methods on ImageNet using a ViT-B/16 classifier. All metrics are averaged over three reference image modes $\mathbf{I}_0$ (black, mean, blur). Top-3 results per metric are highlighted (dark to light).

	Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
Predicted Class	Grad-CAM [55]	0.2858	0.4982	0.2124	0.3346	0.5788	0.2442	84.3339	4.3577	0.014s
	IG [64]	0.1857	0.5544	0.3686	0.2248	0.6347	0.4099	22.9340	33.4353	0.251s
	Trans-Att. [71]	0.1654	0.5833	0.4178	0.2022	0.6654	0.4633	25.6994	30.8467	0.254s
	Bi-Att. [17]	0.1600	0.5958	0.4358	0.1961	0.6789	0.4828	28.4572	28.2614	0.254s
	TIS [27]	0.1420	0.6381	0.4961	0.1731	0.7222	0.5492	**11.2479**	**45.5658**	1.167s
	ViT-CX [69]	0.1750	0.5714	0.3964	0.2149	0.6478	0.4328	12.5472	43.4971	0.860s
	MDA [66]	0.1556	0.6112	0.4556	0.1933	0.6983	0.5050	48.6653	13.2691	13.645s
	LeGrad [10]	0.1666	0.5666	0.4000	0.2036	0.6461	0.4425	20.3850	34.3440	**0.006s**
	AHA (Ours)	0.1381	0.6112	0.4731	0.1714	0.6947	0.5233	15.8713	36.8246	0.016s
	AHA $T = 3$ (Ours)	**0.1215**	**0.6485**	**0.5271**	**0.1491**	**0.7378**	**0.5887**	12.4658	42.2698	0.547s
Ground Truth Class	Grad-CAM [55]	0.2690	0.4638	0.1948	0.3068	0.5244	0.2176	80.4883	7.7366	0.014s
	IG [64]	0.1731	0.5171	0.3440	0.2028	0.5765	0.3737	20.0260	38.5759	0.251s
	Trans-Att. [71]	0.1541	0.5428	0.3887	0.1819	0.6020	0.4200	22.5843	36.0353	0.254s
	Bi-Att. [17]	0.1492	0.5534	0.4042	0.1766	0.6120	0.4354	25.1370	33.5553	0.254s
	TIS [27]	0.1324	0.5966	0.4641	0.1553	0.6595	0.5042	**9.2014**	**50.9796**	1.167s
	ViT-CX [69]	0.1694	0.5231	0.3537	0.2024	0.5748	0.3724	15.2074	42.2452	0.860s
	MDA [66]	0.1461	0.5628	0.4168	0.1756	0.6216	0.4460	44.8870	18.2510	13.645s
	LeGrad [10]	0.1552	0.5280	0.3728	0.1833	0.5859	0.4027	17.6998	39.4492	**0.006s**
	AHA (Ours)	0.1280	0.5657	0.4377	0.1529	0.6222	0.4693	14.0053	41.2158	0.016s
	AHA $T = 3$ (Ours)	**0.1126**	**0.6003**	**0.4877**	**0.1326**	**0.6630**	**0.5303**	10.8509	46.7211	0.547s

on the attribution map. The ADP is then computed as the average percentage drop in the target model's output score between the original image and the masked image. The lower the ADP, the better the attribution map differentiates between the unimportant background and the important foreground. The PIC, on the other hand, is defined as the percentage of cases in which this background-masking leads to an increase in the target model's confidence. Finally, we also report the differences between *Insertion* and *Deletion* AUCs, as well as between *Positive* and *Negative* AUCs, as a measure of the *Faithfulness* [45,54]. To demonstrate that the findings are not mere artefacts of the optimization objective, we provide additional evaluations on MAS [67], ROAD [53] metrics and the ImageNet segmentation benchmark [31] in the supplemental material, which also confirm the performance benefits of AHA.

Table 1 (ViT-B/16) and Table 2 (ViT-B/32) summarize the results of our method compared to a wide range of baselines. Across both analyzed classifiers and both target definitions, AHA without additional refinement steps (T=0) achieves competitive or superior performance to the current state-of-the-art, with particularly strong results on the *Deletion* and *Positive* metrics. We argue that the reason why our *zero-shot* method can compete with, or even outperform, per-sample-optimized methods is because it aligns attribution maps more precisely with the input image. This enables important areas to be localized more accurately than with methods that only operate at patch level. Using only three refinement steps ($T = 3$) at test time, AHA further improves on all metrics, outperforming the baselines in the majority of cases and achieving the best overall *Faithfulness* scores (*Ins.* - *Del.* and *Pos.* - *Neg.*).

Table 2: Quantitative comparison of attribution methods on ImageNet using a ViT-B/32 classifier. All metrics are averaged over three reference image modes $\mathbf{I}_0$ (black, mean, blur). Top-3 results per metric are highlighted (dark to light).

	Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
Predicted Class	Grad-CAM [55]	0.2337	0.5294	0.2956	0.2930	0.6328	0.3398	66.9314	8.5072	0.010s
	IG [64]	0.1897	0.5542	0.3645	0.2426	0.6567	0.4141	14.2831	39.0279	0.157s
	Trans-Att. [71]	0.2024	0.5507	0.3483	0.2567	0.6534	0.3967	17.1553	34.9780	0.157s
	Bi-Att. [17]	0.1887	0.5726	0.3839	0.2416	0.6782	0.4366	17.6613	34.3086	0.158s
	TIS [27]	0.1702	**0.6094**	0.4392	0.2144	0.7181	0.5037	9.7104	48.3937	0.416s
	ViT-CX [69]	0.1982	0.5604	0.3622	0.2525	0.6622	0.4098	9.3272	**48.9950**	0.763s
	MDA [66]	0.1973	0.5939	0.3966	0.2511	0.7043	0.4533	25.2806	26.1375	1.803s
	LeGrad [10]	0.1922	0.5502	0.3580	0.2457	0.6521	0.4065	13.4748	39.4919	**0.005s**
	AHA (Ours)	0.1471	0.5649	0.4178	0.1907	0.6703	0.4796	11.5998	39.7140	0.016s
	AHA $T = 3$ (Ours)	**0.1261**	0.6080	**0.4819**	**0.1600**	**0.7224**	**0.5624**	**7.8339**	47.8573	0.212s
Ground Truth Class	Grad-CAM [55]	0.2185	0.4931	0.2746	0.2647	0.5712	0.3065	63.3098	12.5057	0.010s
	IG [64]	0.1768	0.5157	0.3389	0.2181	0.5910	0.3729	12.7448	43.2325	0.157s
	Trans-Att. [71]	0.1886	0.5126	0.3240	0.2306	0.5883	0.3576	15.2576	39.6312	0.157s
	Bi-Att. [17]	0.1762	0.5316	0.3554	0.2177	0.6071	0.3894	15.7758	38.8666	0.158s
	TIS [27]	0.1589	**0.5691**	0.4102	0.1930	**0.6535**	0.4605	8.0883	**53.2509**	0.416s
	ViT-CX [69]	0.1924	0.5111	0.3187	0.2388	0.5806	0.3418	13.6714	45.4918	0.763s
	MDA [66]	0.1866	0.5443	0.3577	0.2298	0.6191	0.3893	24.5261	29.0981	1.803s
	LeGrad [10]	0.1792	0.5124	0.3331	0.2207	0.5876	0.3669	12.0416	43.7111	**0.005s**
	AHA (Ours)	0.1363	0.5250	0.3887	0.1694	0.6000	0.4306	10.4213	43.8113	0.016s
	AHA $T = 3$ (Ours)	**0.1171**	0.5650	**0.4480**	**0.1420**	0.6497	**0.5077**	**7.0080**	51.5807	0.212s

4.3 Qualitative Evaluation

Figure 1 shows qualitative examples of attribution maps generated by AHA compared to best-performing baselines across the evaluated metrics for the ViT-B/16 classifier. Even without test-time refinement ($T = 0$), AHA produces attribution maps that are more precisely aligned with the input image and better capture object boundaries than the baselines. This is particularly evident for the image classified as *impala*, where our attribution map correctly highlights the animal heads, while the baselines produce more diffuse, blocky maps that also cover the background because of the patch-level attribution. Moreover, in the case of the misclassified *can opener* our method clearly highlights the bottle caps as the most important features for the model's decision, while the baselines produce more noisy maps that are less interpretable. Additional examples for all methods and classifiers are included in the supplementary material.

4.4 Runtime Analysis

The runtime comparison in Tables 1 and 2, measured over 100 iterations on a single NVIDIA RTX 4090, confirms the efficiency advantage of the amortized design. At T=0, AHA requires only a single explainer forward pass and is therefore comparable to the runtime of propagation-based methods like Grad-CAM or LeGrad, while achieving competitive or superior faithfulness compared to methods that are orders of magnitude slower. When refining over T=3 gradient steps, the runtime increases depending on the complexity of the target model, but generally remains below the token/patch perturbation-based methods and significantly below the per-sample metric optimization of MDA. A detailed amortization study is provided in the supplemental material.

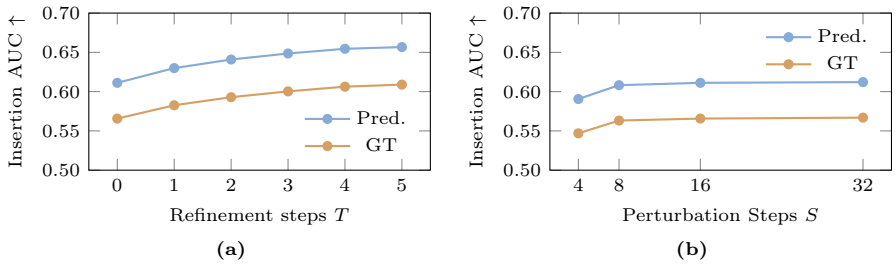


Fig. 3: Effect of (a) test-time refinement steps T and (b) training perturbation steps S on the Insertion AUC. Results are shown for the predicted class (Pred.) and the ground-truth class (GT) on the ImageNet validation set (ViT-B/16).

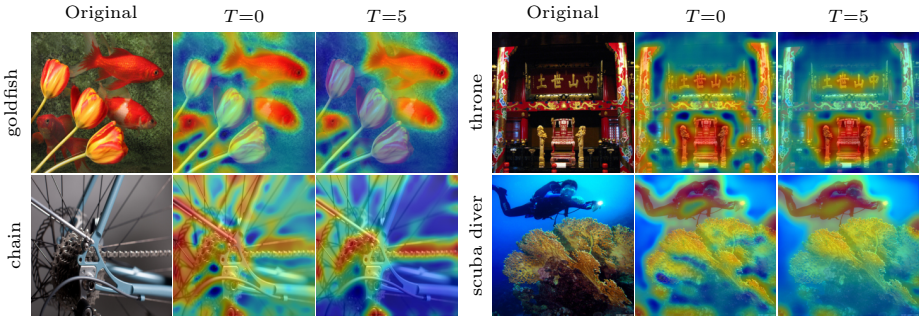


Fig. 4: Exemplary test-time refinement. Attribution maps produced by AHA without (T=0) and with (T=5) test-time refinement for the predicted class using ViT-B/16. Refinement progressively sharpens the heatmaps, concentrating attribution on the relevant object parts while suppressing diffuse background activations.

4.5 Test-Time Refinement Influence

We analyze the effect of applying T gradient steps of test-time refinement on the attribution quality. As shown in Figure 3a, performance improves monotonically with increasing T, with the largest gains concentrated in the first few steps and diminishing returns beyond T=3. Specifically, three refinement steps increase the Insertion AUC from 0.611 to 0.649 (predicted class). The Deletion AUC (where lower is better) follows a similar monotonically improving trend and is reduced from 0.138 to 0.121 for $T = 0 \rightarrow 3$. The qualitative examples in Figure 4 confirm this: refinement progressively sharpens the attribution maps, setting the focus onto the relevant regions and suppressing background activations most clearly visible for the *goldfish* and *scuba diver* examples. The number of refinement steps T thus provides a flexible compute-quality tradeoff, making AHA suitable both for high-throughput settings (T=0) and for scenarios where explanation fidelity is the primary interest.

4.6 Perturbation Steps Influence

During training, the *Deletion* and *Insertion* AUC objectives are approximated by sampling S uniformly spaced perturbation steps instead of evaluating the full sequence. We ablate the effect of this hyperparameter by training models with $S \in \{4, 8, 16, 32\}$. As shown in Figure 3b, the performance improves consistently from S=4 to S=16, after which gains become marginal. This plateau indicates that S=16 already provides a sufficiently dense approximation of the AUC curve.

4.7 Limitations

Despite the state-of-the-art results of our approach, few limitations should be acknowledged. First, each new target model requires a retraining of the explainer. While inference is efficient once trained, this amortization step introduces additional computational cost and limits immediate applicability to arbitrary models. Second, our training objective optimizes approximations of deletion and insertion metrics. While they are widely used to assess attribution faithfulness, they remain proxy measures of explanation quality and have known shortcomings, like sensitivity to the choice of reference image and perturbation strategy (detailed results provided in the supplemental). And third, our method requires access to a representative training dataset for the target model, which may be unavailable for proprietary or closed-source models. However, our optional refinement ($T > 0$) can be employed to resolve such data issues.

5 Conclusion

We introduced AHA, a metric-driven attribution framework that directly optimizes differentiable surrogates of the Deletion and Insertion metrics via a Gumbel–Sinkhorn relaxation of sorting. The proposed approach combines the causal grounding of perturbation-based methods with the efficiency of amortized explainers, producing dense pixel-level attribution maps in a single forward pass. Experiments on ImageNet with Vision Transformers demonstrate improved faithfulness metrics and sharper, boundary-aligned explanations compared to existing methods. These results highlight the potential of learning attribution through direct optimization of evaluation metrics for scalable and faithful model explainability.

Acknowledgements

This work received funding from the AI Ecosystems programme of the Republic of Austria, Federal Ministry of Innovation, Mobility and Infrastructure (BMIMI) under the project DOMINO (923575). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management.

References

1. Abnar, S., Zuidema, W.: Quantifying Attention Flow in Transformers. In: ACL (2020)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity Checks for Saliency Maps. In: NeurIPS (2018)
3. Alshami, E., Agnihotri, S., Schiele, B., Keuper, M.: AIM: Amending Inherent Interpretability via Self-Supervised Masking. In: ICCV (2025)
4. Anders, C.J., Weber, L., Neumann, D., Samek, W., Müller, K.R., Lapuschkin, S.: Finding and Removing Clever Hans: Using Explanation Methods to Debug and Improve Deep Models. Inf. Fusion **77**, 261–295 (Jan 2022)
5. Arya, S., Rao, S., Böhle, M., Schiele, B.: B-cosification: Transforming Deep Neural Networks to be Inherently Interpretable. In: NeurIPS (2024)
6. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE **10**(7), 1–46 (2015)
7. Bhalla, U., Srinivas, S., Lakkaraju, H.: Verifiable Feature Attributions: A Bridge between Post Hoc Explainability and Inherent Interpretability. In: ICML Workshops (2023)
8. Blondel, M., Teboul, O., Berthet, Q., Djolonga, J.: Fast differentiable sorting and ranking. In: ICML (2020)
9. Böhle, M., Fritz, M., Schiele, B.: B-Cos Networks: Alignment Is All We Need for Interpretability. In: CVPR (2022)
10. Bousselham, W., Boggust, A., Chaybouti, S., Strobelt, H., Kuehne, H.: LeGrad: An Explainability Method for Vision Transformers via Feature Formation Sensitivity. In: ICCV (2025)
11. Boyd, A., Bowyer, K.W., Czajka, A.: Human-Aided Saliency Maps Improve Generalization of Deep Learning. In: WACV (2022)
12. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021)
13. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In: WACV (2018)
14. Chefer, H., Gur, S., Wolf, L.: Generic Attention-Model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers. In: ICCV (2021)
15. Chefer, H., Gur, S., Wolf, L.: Transformer Interpretability Beyond Attention Visualization. In: CVPR (2021)
16. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This Looks Like That: Deep Learning for Interpretable Image Recognition. In: NeurIPS (2019)
17. Chen, J., Li, X., Yu, L., Dou, D., Xiong, H.: Beyond Intuition: Rethinking Token Attributions inside Transformers. TMLR (2023)
18. Chen, J., Song, L., Wainwright, M.J., Jordan, M.I.: Learning to explain: An information-theoretic perspective on model interpretation. In: ICML (2018)
19. Chen, R., Liang, S., Li, J., Liu, S., Li, M., Huang, Z., Zhang, H., Cao, X.: Interpreting Object-level Foundation Models via Visual Precision Search. In: CVPR (2025)
20. Chen, R., Zhang, H., Liang, S., Li, J., Cao, X.: Less is More: Fewer Interpretable Region via Submodular Subset Selection. In: ICLR (2024)
21. Covert, I., Kim, C., Lee, S.I., Zou, J., Hashimoto, T.: Stochastic Amortization: A Unified Approach to Accelerate Feature and Data Attribution. In: Advances in Neural Information Processing Systems (2024)

22. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
23. Desai, S., Ramaswamy, H.G.: Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In: WACV (2020)
24. Donnelly, J., Barnett, A.J., Chen, C.: Deformable ProtoPNet: An Interpretable Image Classifier Using Deformable Prototypes. In: CVPR (2022)
25. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
26. Draelos, R.L., Carin, L.: Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. arXiv:2011.08891 (2020)
27. Englebert, A., Stassin, S., Nanfack, G., Mahmoudi, S.A., Siebert, X., Cornu, O., De Vleeschouwer, C.: Explaining Through Transformer Input Sampling. In: ICCV Workshops (2023)
28. Fel, T., Ducoffe, M., Vigouroux, D., Cadène, R., Capelle, M., Nicodème, C., Serre, T.: Don't Lie to Me! Robust and Efficient Explainability With Verified Perturbation Analysis. In: CVPR (2023)
29. Fong, R., Patrick, M., Vedaldi, A.: Understanding Deep Networks via Extremal Perturbations and Smooth Masks. In: ICCV (2019)
30. Fong, R.C., Vedaldi, A.: Interpretable Explanations of Black Boxes by Meaningful Perturbation. In: ICCV (2017)
31. Guillaumin, M., Küttel, D., Ferrari, V.: ImageNet Auto-Annotation with Segmentation Propagation. IJCV **110**(3), 328—-348 (2014)
32. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners. In: CVPR (2022)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
34. Jethani, N., Sudarshan, M., Covert, I.C., Lee, S.I., Ranganath, R.: FastSHAP: Real-Time Shapley Value Estimation. In: ICLR (2022)
35. Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., Wei, Y.: LayerCAM: Exploring Hierarchical Class Activation Maps for Localization. IEEE Transactions on Image Processing **30**, 5875–5888 (2021)
36. Jung, H., Oh, Y.: Towards Better Explanations of Class Activation Mapping. In: ICCV (2021)
37. Kapishnikov, A., Bolukbasi, T., Viegas, F., Terry, M.: XRAI: Better Attributions Through Regions. In: ICCV (2019)
38. Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., Bolukbasi, T.: Guided Integrated Gradients: An Adaptive Path Method for Removing Noise. In: CVPR (2021)
39. Koh, Pang Wei and Nguyen, Thao and Tang, Yew Siang and Mussmann, Stephen and Pierson, Emma and Kim, Been and Liang, Percy: Concept Bottleneck Models. In: ICML (2020)
40. Li, O., Liu, H., Chen, C., Rudin, C.: Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. In: AAAI (2018)
41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: ICCV (2021)
42. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: CVPR (2022)

43. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: NeurIPS (2017)
44. Mena, G., Belanger, D., Linderman, S., Snoek, J.: Learning Latent Permutations with Gumbel-Sinkhorn Networks. In: ICLR (2018)
45. Muzellec, S., Fel, T., Boutin, V., Andéol, L., VanRullen, R., Serre, T.: Saliency strikes back: how filtering out high frequencies improves white-box explanations. In: ICML (2024)
46. Nauta, M., van Bree, R., Seifert, C.: Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In: CVPR (2021)
47. Oikarinen, T., Das, S., Nguyen, L.M., Weng, T.W.: Label-free Concept Bottleneck Models. In: ICLR (2023)
48. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized Input Sampling for Explanation of Black-box Models. In: BMVC (2018)
49. Prillo, S., Eisenschlos, J.M.: SoftSort: A Continuous Relaxation for the argsort Operator. In: ICML (2020)
50. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: ICML (2021)
51. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision Transformers for Dense Prediction. In: ICCV (2021)
52. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: KDD (2016)
53. Rong, Y., Leemann, T., Borisov, V., Kasneci, G., Kasneci, E.: A Consistent and Efficient Evaluation Strategy for Attribution Methods. In: ICML (2022)
54. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a Deep Neural Network has learned. IEEE Transactions on Neural Networks and Learning Systems **28**(11), 2660–2673 (2016)
55. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: ICCV (2017)
56. Shrikumar, A., Greenside, P., Kundaje, A.: Learning Important Features Through Propagating Activation Differences. In: ICML (2017)
57. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. arXiv:2508.10104 (2025)
58. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 (2013)
59. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967)
60. Situ, X., Zukerman, I., Paris, C., Maruf, S., Haffari, G.: Learning to Explain: Generating Stable Explanations Fast". In: ACL (2021)
61. Sixt, L., Granz, M., Landgraf, T.: When explanations lie: why many modified BP attributions fail. In: ICML (2020)
62. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for Simplicity: The All Convolutional Net. In: ICLR (2015)
63. Srinivas, S., Fleuret, F.: Full-Gradient Representation for Neural Network Visualization. In: NeurIPS (2019)
64. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic Attribution for Deep Networks. In: ICML (2017)

65. Tran, M., Lahiani, A., Dicente Cid, Y., Boxberg, M., Lienemann, P., Matek, C., Wagner, S.J., Theis, F.J., Klaiman, E., Peng, T.: B-Cos Aligned Transformers Learn Human-Interpretable Features. In: MICCAI (2023)
66. Walker, C., Jha, S.K., Ewetz, R.: Metric-Driven Attributions for Vision Transformers. In: ICLR (2025)
67. Walker, C., Simon, D., Chen, K., Ewetz, R.: Attribution Quality Metrics with Magnitude Alignment. In: IJCAI (2024)
68. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-CAM: Score-weighted visual explanations for convolutional neural networks. In: CVPR Workshops (2020)
69. Xie, W., Li, X.H., Cao, C.C., Zhang, N.L.: ViT-CX: Causal Explanation of Vision Transformers. In: IJCAI (2023)
70. Xu, S., Venugopalan, S., Sundararajan, M.: Attribution in Scale and Space. In: CVPR (2020)
71. Yuan, T., Li, X., Xiong, H., Cao, H., Dou, D.: Explaining Information Flow Inside Vision Transformers Using Markov Chain. In: NeurIPS Workshop XAI4Debugging (2021)
72. Zeiler, M.D., Fergus, R.: Visualizing and Understanding Convolutional Networks. In: ECCV (2014)
73. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-Down Neural Attention by Excitation Backprop. IJCV **126**(10), 1084–1102 (2018)
74. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. In: CVPR (2016)
75. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In: ICLR (2017)

A Appendix

This appendix complements the main paper with additional implementation details, evaluations, and qualitative results. We first detail the architecture of the explainer (Sec. A.1), analyze its training cost and amortization (Sec. A.2), and study out-of-distribution data via test-time refinement (Sec. A.3). We then demonstrate the architecture-agnostic applicability by explaining a convolutional classifier (Sec. A.4), evaluate AHA under the MDA protocol (Sec. A.5), on complementary metrics (Sec. A.6), and on the ImageNet segmentation benchmark (Sec. A.7), and assess cross-architecture explainer transfer (Sec. A.8). Finally, we ablate the remaining hyperparameters (Sec. A.9), report detailed per-reference-image results (Sec. A.10), show the full AUC curves (Sec. A.11), and present qualitative results (Sec. A.12).

A.1 Explainer Architecture Details

Our attribution model $\Phi_{f,\theta}$ follows an encoder-decoder design. The encoder is a frozen DINOv3 [57] ViT-L/16 backbone. We extract intermediate features from four evenly spaced transformer blocks (layers 4, 11, 17, 23), each producing spatial feature maps of dimension 1024. The decoder follows the Dense Prediction Transformer (DPT) [51] design. At each of the four scales, the patch-level features are concatenated with a learned class embedding $\mathbf{e}_t \in \mathbb{R}^{1024}$, obtained from an embedding layer that maps the target class index t to a dense vector, and projected back to 256 channels via a linear layer followed by GELU activation. This enables the model to produce different attribution maps depending on the target class. The four resulting feature maps are progressively fused through residual convolution units and bilinear upsampling, yielding a single 128-channel feature map at the highest resolution. A lightweight refinement head consisting of a depthwise-separable convolution followed by an additional 3×3 convolution, each with GELU activation and batch normalization, maps this representation through a final convolutional layer to the single-channel attribution map $\mathbf{A} \in \mathbb{R}^{H \times W}$, where H and W are the height and width of the input image, respectively. During training, only the decoder parameters and the class embedding are optimized while the DINOv3 backbone remains entirely frozen.

A.2 Training Cost and Amortization

For each target model, the explainer requires a one-time training. After that, it efficiently generates attributions during inference, allowing its training costs to be amortized across the generated attributions. For ViT-B/16, training on ImageNet with S=16 perturbation steps takes ∼4.5h on four NVIDIA RTX 6000 GPUs. Table 3 reports the resulting *break-even points*, *i.e.* the number of attributions after which the total runtime of AHA (training plus inference) undercuts that of a more time-consuming baseline. Against the per-sample optimization of

Table 3: Training cost amortization. Break-even point (number of generated attributions) after which the total runtime of AHA, including the one-time training, falls below that of the baseline. Results are shown for ViT-B/16 with S=16 and S=8 perturbation steps during training and with T=0 or T=3 test-time refinement steps at inference. Lower is better.

	S=16 (train ∼4.5h)		S=8 (train ∼2h)	
Baseline	T=0	T=3	T=0	T=3
ViT-CX	19.19k	51.76k	8.53k	23.00k
TIS	14.07k	26.13k	6.25k	11.61k
MDA	1.19k	1.24k	0.53k	0.55k

MDA [66] this happens after ∼1k generated attribution maps, against TIS [27] after ∼14k and for ViT-CX [69] after ∼19k samples. Reducing the perturbation steps to S=8 halves the training time to ∼2h at only a small quality drop (see Fig. 3(b) main paper), lowering all break-even points accordingly.

A.3 Out-of-Distribution Attribution via Test-Time Refinement

To evaluate whether test-time refinement can detect spurious correlations in previously unseen models or data, we construct a *controlled shortcut-learning scenario*: A ViT-B/16 is fine-tuned on ImageNet for one epoch with a small white circle injected into every training image of the *sunglasses* class. We choose this class for its low baseline top-1 accuracy, which makes the shortcut effect particularly pronounced. After fine-tuning, the accuracy on circle-augmented *sunglasses* images rises to 100%. Crucially, the AHA explainer was trained on the original, unmodified ViT-B/16 and has never encountered the white-circle artifact, placing it in an out-of-distribution setting. At inference, the explainer generates an initial attribution map. During refinement steps $T \in \{1, \ldots, 5\}$, the fine-tuned model serves as the target model f for backpropagation-based updates. As shown in Figure 5, at T=0 the explainer highlights generic class-discriminative features. As T increases, the refinement progressively shifts the attribution map toward the white circle, revealing the shortcut the fine-tuned model relies on.

A.4 Architecture-Agnostic Applicability

To demonstrate the universal applicability of AHA beyond transformer-based classifiers, we apply our approach without any adaptions also to a ConvNeXt Small [42] classifier and compare against established attribution methods applicable to convolutional architectures. Table 4 summarizes the results. Also in this setting, AHA achieves the best overall faithfulness. Without refinement (T=0), our method already reduces the *Deletion* significantly compared to the strongest

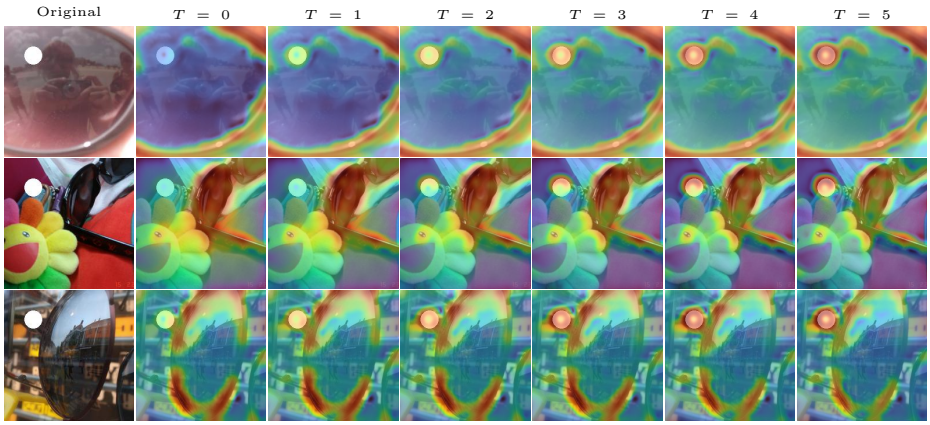


Fig. 5: Out-of-distribution shortcut detection. A ViT-B/16 is fine-tuned with a white circle injected into every *sunglasses* training image. Using an AHA explainer trained on the *clean* model, test-time refinement progressively reveals the learned shortcut: at T=0 the attribution highlights generic class features, while increasing T reveals the white circle as important feature for the fine-tuned model's decision.

baseline GradCAM/HiResCAM (0.2748 → 0.1741, predicted class), indicating that the most-attributed regions are highly relevant for the classifier's decision. The ADP also drops substantially, confirming a tighter foreground isolation. With three refinement steps (T=3), AHA further improves across all metrics and achieves the highest scores on 6 out of 8 metrics for both target class definitions, including the composite faithfulness scores (*Ins.* - *Del.* and *Neg.* - *Pos.*). The qualitative examples in Figure 9 confirm our findings. AHA produces dense, boundary-aligned attribution maps that closely follow the object silhouette, whereas the CAM-based baselines yield coarser heatmaps that often spread into the background. This is particularly visible for the *German shepherd* and *whistle* example, where our method clearly aligns with the object contours.

A.5 Evaluation under the MDA Protocol

For a fair comparison with MDA [66], we additionally evaluate AHA on the exact image subset and protocol of their official codebase. Unlike our standard evaluation on the full ImageNet validation set (50K images), MDA applies several filters to select a 5,000 image subset (5 high-confidence images per class). Furthermore, while our protocol averages metrics over three reference images $\mathbf{I}_0$ (black, mean, blur), MDA suggests using a single per-image adaptive blur for Insertion and a black image for Deletion. As shown in Table 5, AHA performs favorably across all metrics under this protocol, with substantially lower runtime compared to MDA.

Table 4: Quantitative comparison of attribution methods on ImageNet using a ConvNeXt Small classifier [42]. All metrics are averaged over three reference image modes $\mathbf{I}_0$ (black, mean, blur). Top-3 results per metric are highlighted (dark to light).

	Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
Pred. Class	ScoreCAM [68]	0.3779	0.5425	0.1646	0.5348	0.7367	0.2019	56.0312	5.8985	0.821s
	AblationCAM [23]	0.3340	0.5890	0.2550	0.4753	0.7877	0.3124	48.9473	4.0993	0.742s
	GradCAM++ [13]	0.3159	0.6008	0.2848	0.4635	0.8231	0.3597	50.9349	4.3820	0.019s
	HiResCAM [26]	0.2748	0.6267	0.3520	0.3908	0.8314	0.4406	41.5140	5.4472	0.019s
	GradCAM [55]	0.2748	**0.6267**	0.3520	0.3908	**0.8314**	0.4406	41.5133	5.4453	**0.004s**
	AHA (Ours)	0.1741	0.5911	0.4170	0.2408	0.7616	0.5208	23.6857	6.0192	0.005s
	AHA $T = 3$ (Ours)	**0.1576**	0.6079	**0.4503**	**0.2150**	0.7842	**0.5692**	**22.1279**	**7.1759**	0.313s
GT Class	ScoreCAM [68]	0.3520	0.5001	0.1481	0.4856	0.6595	0.1739	52.7301	11.0598	0.821s
	AblationCAM [23]	0.3117	0.5429	0.2313	0.4336	0.7023	0.2688	44.5182	10.6520	0.742s
	GradCAM++ [13]	0.2959	0.5533	0.2573	0.4258	0.7331	0.3074	45.5688	11.9073	0.019s
	HiResCAM [26]	0.2578	0.5771	0.3193	0.3600	0.7415	0.3815	37.0658	13.1477	0.019s
	GradCAM [55]	0.2578	**0.5771**	0.3193	0.3600	**0.7415**	0.3815	37.0650	13.1467	**0.004s**
	AHA (Ours)	0.1624	0.5467	0.3843	0.2191	0.6828	0.4637	21.2367	12.6424	0.005s
	AHA $T = 3$ (Ours)	**0.1472**	0.5613	**0.4141**	**0.1957**	0.7028	**0.5071**	**19.7873**	**14.0637**	0.313s

Table 5: Quantitative comparison using the MDA evaluation framework [66] for a ViT-B/16 classifier on their filtered ImageNet subset (5,000 images). Top-3 results per metric are highlighted (dark to light).

Method	Deletion↓	Insertion↑	Insertion - Deletion↑	Runtime
Grad-CAM [55]	0.241	0.737	0.496	**0.014s**
IG [64]	0.222	0.741	0.518	0.251s
ViT-CX [69]	0.236	0.722	0.486	0.860s
Trans-Att. [71]	0.228	0.748	0.520	0.254s
Bi-Att. [17]	0.218	0.760	0.542	0.254s
TIS [27]	0.196	0.761	0.565	1.167s
MDA [66]	0.232	0.856	0.624	13.645s
AHA (Ours)	0.147	0.862	0.716	0.016s
AHA $T = 3$ (Ours)	**0.131**	**0.886**	**0.756**	0.547s

A.6 Complementary Faithfulness Metrics

To probe whether the faithfulness of AHA generalizes beyond the rank-based *Deletion* and *Insertion* training objective, we evaluate two complementary metrics on the same 5,000-image subset and protocol as MDA [66] (Section A.5) for the ViT-B/16 classifier. Table 6 shows the magnitude-sensitive MAS [67], which weights the perturbation by the attribution magnitudes rather than by their rank alone, and the debiased ROAD [53], which replaces removed regions by a noisy linear imputation to reduce information leakage from the mask shape. $\text{AHA}_{T=0}$ remains competitive with the per-sample-optimized MDA on the magnitude-based MAS and achieves the best scores on the debiased ROAD metric.

A.7 Localization via ImageNet Segmentation

To demonstrate that AHA learns spatially meaningful, object-aligned attribution maps, we evaluate it on the ImageNet segmentation benchmark [31], treating the thresholded attribution map as a foreground prediction and comparing it against

Table 6: Complementary faithfulness metrics. Magnitude-based MAS [67] and debiased ROAD [53] on the 5,000-image MDA subset for the ViT-B/16 classifier. Top-3 results per metric are highlighted (dark to light).

	MAS			ROAD		
Method	Ins.↑	Del.↓	Diff.↑	LeRF↑	MoRF↓	Diff.↑
ViT-CX [69]	0.582	0.408	0.174	0.664	0.201	0.463
TIS [27]	0.615	0.385	0.230	0.712	0.169	0.543
MDA [66]	**0.775**	**0.279**	**0.497**	0.687	0.189	0.498
AHA (T=0)	0.761	0.290	0.471	**0.715**	**0.160**	**0.555**

Table 7: ImageNet segmentation. Localization performance on the ImageNet segmentation benchmark [31] for the ViT-B/32 classifier, following the MDA [66] setting. Top-3 results per metric are highlighted (dark to light).

Method	Mean Avg. Precision↑	Intersection-over-Union↑	F1 Score↑
ViT-CX [69]	0.678	0.535	0.428
TIS [27]	0.705	0.586	0.460
MDA [66]	**0.796**	**0.702**	**0.487**
AHA (T=0)	0.790	0.667	0.483

the ground-truth object masks [15]. We follow the same setting as MDA [66] for the ViT-B/32 classifier and report the mean average precision (MAP), the pixel-wise intersection-over-union (IoU), and the F1 score in Table 7. $\text{AHA}_{T=0}$ outperforms ViT-CX [69] and TIS [27] across all three metrics, achieving comparable performance to the per-sample-optimized MDA. This confirms that the attribution maps generated by AHA reflect object localizations and are not an artifact of optimizing the *Deletion* and *Insertion* objectives.

A.8 Cross-Architecture Explainer Transfer

Beyond applying AHA to different architectures by training a dedicated explainer for each (Section A.4), we further investigate whether a trained explainer transfers to a related architecture without retraining. This is plausible for correctly-classifying architectures, as they tend to rely on similar object evidence. In Table 8 we evaluate, on correctly classified ImageNet samples, how the faithfulness metrics change when an explainer optimized for one ViT is applied to the other, using the per-sample-optimized MDA [66] as a reference point. $\text{AHA}_{T=0}$ exhibits robust zero-shot transfer, with consistently smaller drops than MDA in both directions. Moreover, the low-cost T=3 refinement adapts the transferred explainer to the target architecture and recovers most of the remaining model-specific gap, in some cases even surpassing the native result. This indicates that trained AHA explainers provide strong initializations for related architectures, substantially reducing the need for full retraining.

Table 8: Cross-architecture transfer. RISE [48] and MAS [67] faithfulness metrics (*Insertion*−*Deletion* difference) on correctly classified ImageNet samples. Gray rows are the native reference, where the explainer is optimized for the same architecture it is evaluated on. In the remaining transfer rows, an explainer trained for a different model was used.

	Method	Optimized for		Evaluated on	RISE Diff.↑	RISE Δ	MAS Diff.↑	MAS Δ
ViT-B/16	**MDA**	**B/16**	→	**B/16**	**0.624**	–	**0.497**	–
	MDA	B/32	→	B/16	0.473	**-24.2%**	0.336	**-32.4%**
	AHA$_{T=0}$	**B/16**	→	**B/16**	**0.715**	–	**0.471**	–
	AHA$_{T=0}$	B/32	→	B/16	0.650	**-9.1%**	0.417	**-11.5%**
	AHA$_{T=3}$	B/32	→	B/16	0.739	**+3.3%**	0.427	**-9.3%**
ViT-B/32	**MDA**	**B/32**	→	**B/32**	**0.708**	–	**0.616**	–
	MDA	B/16	→	B/32	0.612	**-13.6%**	0.509	**-17.4%**
	AHA$_{T=0}$	**B/32**	→	**B/32**	**0.723**	–	**0.589**	–
	AHA$_{T=0}$	B/16	→	B/32	0.683	**-5.5%**	0.544	**-7.6%**
	AHA$_{T=3}$	B/16	→	B/32	0.737	**+1.9%**	0.590	**+0.1%**

Table 9: Hyperparameter ablations. Effect of the training grid size G and the sorting temperature τ on faithfulness (RISE and MAS *Insertion*−*Deletion* difference, higher is better) for the ViT-B/16 classifier on correctly classified samples. The configuration used throughout the paper is marked *(used)* and shown in bold.

Grid size G	RISE Diff.↑	MAS Diff.↑	Temp. τ	RISE Diff.↑	MAS Diff.↑
$G\sim\mathcal{U}(7,28)$ *(used)*	**0.715**	**0.471**	τ=1.0 *(used)*	**0.715**	**0.471**
Fixed G=7	0.671	0.427	τ=0.5	0.703	0.462
Fixed G=14	0.698	0.453	τ=1.5	0.710	0.469
Fixed G=28	0.710	0.470	τ=2.0	0.708	0.467

A.9 Additional Hyperparameter Ablations

Complementing the test-time refinement (T) and perturbation-step (S) ablations in the main paper, we ablate the remaining training hyperparameters in Table 9. The grid size G partitions the image into K=G^2 regions over which the region-level training perturbations are defined. Although AHA stays largely stable w.r.t. the choice of G, we sample it uniformly from $\mathcal{U}(7,28)$ at each training iteration anyway, which avoids overfitting to a single perturbation granularity and yields the best faithfulness. We further emphasize that the grid only defines the region-level perturbations during training: at inference, AHA always produces a dense, pixel-level attribution map. The sorting temperature τ controls the smoothness of the Gumbel-Sinkhorn relaxation. Here the faithfulness metrics vary only marginally across the tested values, with τ=1.0 performing best.

A.10 Detailed Main Results

Tables 10 to 12 provide the full per-reference-image results for all evaluated classifiers (ViT-B/16, ViT-B/32, ConvNeXt Small). While the main paper re-

ports metrics averaged over the three reference image modes $\mathbf{I}_0$ (black, mean, blur), these tables list the individual results for each mode separately, allowing for a more fine-grained analysis of the sensitivity to the choice of reference image.

A.11 AUC Plots

Figure 6 shows the full Deletion and Insertion AUC curves for ViT-B/16 and ViT-B/32, averaged over all reference images. The curves illustrate how the target model's confidence changes as pixels are progressively removed (Deletion) or revealed (Insertion) in order of their attributed importance. AHA consistently achieves steeper Deletion curves and steeper Insertion curves than the baselines, confirming that its ranking of pixel importance aligns more closely with the classifier's actual reliance on image regions.

A.12 Qualitative Results

Figures 7 to 9 present qualitative attribution maps for all evaluated classifiers (ViT-B/16, ViT-B/32, ConvNeXt Small) across a diverse set of ImageNet validation samples, including both correctly and incorrectly classified images. Across all architectures, AHA produces sharper, more boundary-aligned attribution maps compared to the baselines, which tend to produce coarser, patch- or region-level activations.

Table 10: Quantitative comparison of attribution methods on ImageNet using a ViT-B/16 classifier [25] for all reference image modes $\mathbf{I}_0$ (black image, mean-colored image, blurred original image). Top-3 results per metric are highlighted (dark to light).

		Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
$\mathbf{I}_0 \equiv$ Black Image	Predicted Class	Grad-CAM [55]	0.2542	0.4562	0.2020	0.3024	0.5336	0.2312	90.2340	2.6849	0.014s
		IG [64]	0.1555	0.5063	0.3508	0.1899	0.5857	0.3958	26.4955	30.3434	0.251s
		Trans-Att. [71]	0.1402	0.5380	0.3978	0.1726	0.6189	0.4463	29.7100	27.5834	0.254s
		Bi-Att. [17]	0.1355	0.5533	0.4178	0.1669	0.6348	0.4678	32.5737	25.3995	0.254s
		TIS [27]	0.1205	0.5976	0.4771	0.1474	0.6807	0.5332	**14.1945**	**42.2192**	1.167s
		ViT-CX [69]	0.1488	0.5269	0.3781	0.1845	0.6019	0.4174	17.1795	37.5492	0.860s
		MDA [66]	0.1336	0.5769	0.4433	0.1669	0.6617	0.4948	59.7938	8.3518	13.645s
		LeGrad [10]	0.1390	0.5189	0.3799	0.1715	0.5970	0.4254	24.2647	31.1134	**0.006s**
		AHA (Ours)	0.1195	0.5691	0.4496	0.1489	0.6502	0.5013	18.1877	35.0353	0.016s
		AHA $T = 3$ (Ours)	**0.1079**	**0.6005**	**0.4926**	**0.1336**	**0.6883**	**0.5547**	14.8003	39.8492	0.547s
	Ground Truth Class	Grad-CAM [55]	0.2398	0.4252	0.1854	0.2780	0.4845	0.2065	86.7297	5.5450	0.014s
		IG [64]	0.1449	0.4744	0.3295	0.1711	0.5351	0.3641	23.2480	35.6353	0.251s
		Trans-Att. [71]	0.1304	0.5032	0.3728	0.1548	0.5645	0.4097	26.3366	32.8113	0.254s
		Bi-Att. [17]	0.1261	0.5165	0.3904	0.1497	0.5771	0.4274	28.9729	30.6954	0.254s
		TIS [27]	0.1123	**0.5600**	0.4478	0.1317	**0.6241**	0.4924	**11.7329**	**47.9150**	1.167s
		ViT-CX [69]	0.1436	0.4852	0.3416	0.1730	0.5384	0.3653	18.1427	38.4452	0.860s
		MDA [66]	0.1251	0.5332	0.4081	0.1508	0.5928	0.4420	55.2688	13.5977	13.645s
		LeGrad [10]	0.1293	0.4858	0.3565	0.1538	0.5449	0.3912	21.1425	36.3753	**0.006s**
		AHA (Ours)	0.1107	0.5288	0.4181	0.1325	0.5865	0.4540	16.1274	39.5132	0.016s
		AHA $T = 3$ (Ours)	**0.1000**	0.5577	**0.4577**	**0.1186**	0.6218	**0.5032**	12.9713	44.3711	0.547s
$\mathbf{I}_0 \equiv$ Mean-colored Image	Predicted Class	Grad-CAM [55]	0.2802	0.4881	0.2079	0.3269	0.5684	0.2415	86.3553	4.0605	0.014s
		IG [64]	0.1751	0.5429	0.3678	0.2123	0.6229	0.4106	23.2566	33.1053	0.251s
		Trans-Att. [71]	0.1551	0.5730	0.4179	0.1893	0.6547	0.4654	25.4997	30.9174	0.254s
		Bi-Att. [17]	0.1506	0.5853	0.4347	0.1844	0.6676	0.4832	28.1998	28.2094	0.254s
		TIS [27]	0.1331	0.6283	0.4952	0.1619	0.7117	0.5497	11.5112	44.4791	1.167s
		ViT-CX [69]	0.1638	0.5624	0.3987	0.2006	0.6384	0.4377	**10.7794**	**45.6811**	0.860s
		MDA [66]	0.1410	0.6006	0.4596	0.1754	0.6869	0.5115	49.9938	12.5057	13.645s
		LeGrad [10]	0.1554	0.5565	0.4011	0.1902	0.6358	0.4456	20.4968	33.7513	**0.006s**
		AHA (Ours)	0.1327	0.5985	0.4658	0.1647	0.6808	0.5161	16.3669	36.0993	0.016s
		AHA $T = 3$ (Ours)	**0.1177**	**0.6356**	**0.5179**	**0.1446**	**0.7249**	**0.5803**	12.6456	41.8452	0.547s
	Ground Truth Class	Grad-CAM [55]	0.2638	0.4545	0.1907	0.2999	0.5156	0.2157	82.4823	7.3558	0.014s
		IG [64]	0.1631	0.5068	0.3437	0.1910	0.5667	0.3757	20.2897	38.2752	0.251s
		Trans-Att. [71]	0.1443	0.5337	0.3894	0.1698	0.5935	0.4237	22.3643	36.1313	0.254s
		Bi-Att. [17]	0.1402	0.5443	0.4041	0.1655	0.6034	0.4379	24.8366	33.6413	0.254s
		TIS [27]	0.1240	0.5877	0.4638	0.1448	0.6511	0.5063	**9.3110**	**49.9430**	1.167s
		ViT-CX [69]	0.1585	0.5150	0.3565	0.1889	0.5670	0.3781	14.5135	43.0011	0.860s
		MDA [66]	0.1322	0.5537	0.4215	0.1590	0.6132	0.4542	46.0686	17.5416	13.645s
		LeGrad [10]	0.1447	0.5191	0.3744	0.1710	0.5775	0.4065	17.7684	38.9052	**0.006s**
		AHA (Ours)	0.1226	0.5552	0.4326	0.1461	0.6126	0.4664	14.3323	40.5832	0.016s
		AHA $T = 3$ (Ours)	**0.1089**	**0.5895**	**0.4806**	**0.1282**	**0.6540**	**0.5259**	10.8600	46.6231	0.547s
$\mathbf{I}_0 \equiv$ Blurred Original Image	Predicted Class	Grad-CAM [55]	0.3230	0.5503	0.2273	0.3744	0.6343	0.2600	76.4122	6.3277	0.014s
		IG [64]	0.2266	0.6139	0.3872	0.2722	0.6956	0.4234	19.0499	36.8573	0.251s
		Trans-Att. [71]	0.2011	0.6388	0.4377	0.2447	0.7227	0.4781	21.8884	34.0393	0.254s
		Bi-Att. [17]	0.1940	0.6489	0.4549	0.2370	0.7344	0.4974	24.5980	31.1754	0.254s
		TIS [27]	0.1725	0.6884	0.5159	0.2098	0.7744	0.5645	**8.0380**	**49.9990**	1.167s
		ViT-CX [69]	0.2125	0.6249	0.4124	0.2596	0.7030	0.4434	9.6828	47.2611	0.860s
		MDA [66]	0.1923	0.6561	0.4638	0.2376	0.7463	0.5087	36.2084	18.9496	13.645s
		LeGrad [10]	0.2053	0.6243	0.4190	0.2492	0.7057	0.4565	16.3933	38.1672	**0.006s**
		AHA (Ours)	0.1621	0.6660	0.5039	0.2005	0.7531	0.5526	13.0591	39.3392	0.016s
		AHA $T = 3$ (Ours)	**0.1387**	**0.7095**	**0.5708**	**0.1691**	**0.8002**	**0.6312**	9.9514	45.1151	0.547s
	Ground Truth Class	Grad-CAM [55]	0.3034	0.5116	0.2082	0.3424	0.5732	0.2307	72.2530	10.3090	0.014s
		IG [64]	0.2114	0.5702	0.3588	0.2463	0.6276	0.3813	16.5402	41.8172	0.251s
		Trans-Att. [71]	0.1876	0.5914	0.4038	0.2212	0.6480	0.4268	19.0520	39.1632	0.254s
		Bi-Att. [17]	0.1812	0.5994	0.4182	0.2146	0.6556	0.4410	21.6014	36.3293	0.254s
		TIS [27]	0.1611	0.6419	0.4808	0.1895	0.7032	0.5137	**6.5602**	**55.0809**	1.167s
		ViT-CX [69]	0.2060	0.5692	0.3632	0.2451	0.6191	0.3739	12.9660	45.2891	0.860s
		MDA [66]	0.1808	0.6016	0.4207	0.2169	0.6587	0.4418	33.3235	23.6135	13.645s
		LeGrad [10]	0.1916	0.5792	0.3876	0.2250	0.6353	0.4103	14.1884	43.0671	**0.006s**
		AHA (Ours)	0.1507	0.6131	0.4624	0.1800	0.6676	0.4876	11.5561	43.5511	0.016s
		AHA $T = 3$ (Ours)	**0.1290**	**0.6537**	**0.5247**	**0.1512**	**0.7131**	**0.5620**	8.7216	49.1690	0.547s

Table 11: Quantitative comparison of attribution methods on ImageNet using a ViT-B/32 classifier [25] for all reference image modes $\mathbf{I}_0$ (black image, mean-colored image, blurred original image). Top-3 results per metric are highlighted (dark to light).

		Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
$\mathbf{I}_0 \equiv$ Black Image	Predicted Class	Grad-CAM [55]	0.1865	0.4705	0.2840	0.2365	0.5688	0.3323	70.1550	7.6858	0.010s
		IG [64]	0.1483	0.4963	0.3480	0.1905	0.5940	0.4035	16.2898	37.6572	0.157s
		Trans-Att. [71]	0.1623	0.4923	0.3300	0.2064	0.5902	0.3838	19.2848	33.7333	0.157s
		Bi-Att. [17]	0.1497	0.5206	0.3709	0.1914	0.6214	0.4300	19.6633	33.5833	0.158s
		TIS [27]	0.1368	**0.5555**	0.4186	0.1727	**0.6607**	0.4880	11.6713	**46.1231**	0.416s
		ViT-CX [69]	0.1601	0.5023	0.3422	0.2049	0.5993	0.3944	12.4364	44.0451	0.763s
		MDA [66]	0.1604	0.5502	0.3897	0.2042	0.6555	0.4513	29.0651	23.3835	1.803s
		LeGrad [10]	0.1517	0.4890	0.3373	0.1952	0.5859	0.3907	15.4655	37.8712	**0.005s**
		AHA (Ours)	0.1197	0.4947	0.3750	0.1548	0.5951	0.4402	13.6634	37.6672	0.016s
		AHA $T = 3$ (Ours)	**0.1062**	0.5266	**0.4204**	**0.1359**	0.6379	**0.5020**	**9.6917**	45.0871	0.212s
	Ground Truth Class	Grad-CAM [55]	0.1745	0.4397	0.2652	0.2137	0.5163	0.3027	66.8279	11.2838	0.010s
		IG [64]	0.1381	0.4635	0.3254	0.1704	0.5382	0.3679	14.6112	41.6852	0.157s
		Trans-Att. [71]	0.1512	0.4600	0.3089	0.1847	0.5356	0.3508	17.2930	38.1932	0.157s
		Bi-Att. [17]	0.1397	0.4851	0.3455	0.1717	0.5605	0.3888	17.7274	37.8892	0.158s
		TIS [27]	0.1277	**0.5196**	0.3920	0.1545	**0.6035**	0.4490	9.9048	**51.1050**	0.416s
		ViT-CX [69]	0.1545	0.4611	0.3066	0.1920	0.5311	0.3391	15.5775	42.4012	0.763s
		MDA [66]	0.1514	0.5057	0.3543	0.1858	0.5801	0.3943	27.6928	26.7615	1.803s
		LeGrad [10]	0.1414	0.4571	0.3157	0.1748	0.5316	0.3568	13.9138	41.9092	**0.005s**
		AHA (Ours)	0.1109	0.4615	0.3506	0.1370	0.5374	0.4004	12.4512	41.6292	0.016s
		AHA $T = 3$ (Ours)	**0.0986**	0.4910	**0.3925**	**0.1200**	0.5774	**0.4573**	**8.7874**	48.7930	0.212s
$\mathbf{I}_0 \equiv$ Mean-colored Image	Predicted Class	Grad-CAM [55]	0.2322	0.5177	0.2855	0.2908	0.6226	0.3318	66.6890	7.8998	0.010s
		IG [64]	0.1831	0.5426	0.3594	0.2338	0.6457	0.4120	14.7945	36.8733	0.157s
		Trans-Att. [71]	0.1941	0.5407	0.3466	0.2459	0.6440	0.3981	17.7665	31.9514	0.157s
		Bi-Att. [17]	0.1830	0.5625	0.3796	0.2336	0.6690	0.4354	18.1669	31.2494	0.158s
		TIS [27]	0.1634	**0.6000**	0.4366	0.2049	0.7106	0.5057	9.9382	46.1071	0.416s
		ViT-CX [69]	0.1920	0.5507	0.3587	0.2443	0.6542	0.4099	**7.9095**	**50.5990**	0.763s
		MDA [66]	0.1898	0.5811	0.3914	0.2408	0.6927	0.4519	25.6178	24.6295	1.803s
		LeGrad [10]	0.1849	0.5386	0.3537	0.2362	0.6413	0.4051	13.8928	37.3753	**0.005s**
		AHA (Ours)	0.1396	0.5533	0.4137	0.1804	0.6587	0.4783	12.1135	37.8692	0.016s
		AHA $T = 3$ (Ours)	**0.1205**	0.5988	**0.4783**	**0.1531**	**0.7155**	**0.5624**	7.9805	46.7151	0.212s
	Ground Truth Class	Grad-CAM [55]	0.2171	0.4823	0.2652	0.2627	0.5626	0.2999	62.7745	12.2258	0.010s
		IG [64]	0.1705	0.5054	0.3349	0.2099	0.5826	0.3728	13.0733	41.3392	0.157s
		Trans-Att. [71]	0.1808	0.5037	0.3228	0.2208	0.5810	0.3602	15.6204	36.8793	0.157s
		Bi-Att. [17]	0.1707	0.5229	0.3522	0.2100	0.6008	0.3908	16.0197	36.0913	0.158s
		TIS [27]	0.1525	**0.5610**	0.4085	0.1843	**0.6482**	0.4639	8.1599	**51.2290**	0.416s
		ViT-CX [69]	0.1863	0.5028	0.3164	0.2313	0.5746	0.3433	13.1598	45.7631	0.763s
		MDA [66]	0.1795	0.5336	0.3541	0.2201	0.6109	0.3908	24.7162	27.8754	1.803s
		LeGrad [10]	0.1723	0.5022	0.3299	0.2120	0.5792	0.3672	12.2789	41.8652	**0.005s**
		AHA (Ours)	0.1290	0.5155	0.3865	0.1595	0.5933	0.4337	10.6786	42.2772	0.016s
		AHA $T = 3$ (Ours)	**0.1117**	0.5579	**0.4462**	**0.1354**	0.6472	**0.5118**	**6.9561**	50.9230	0.212s
$\mathbf{I}_0 \equiv$ Blurred Original Image	Predicted Class	Grad-CAM [55]	0.2825	0.5999	0.3174	0.3516	0.7070	0.3554	63.9501	9.9358	0.010s
		IG [64]	0.2378	0.6237	0.3859	0.3036	0.7305	0.4268	11.7650	42.5531	0.157s
		Trans-Att. [71]	0.2508	0.6191	0.3683	0.3178	0.7259	0.4081	14.4145	39.2492	0.157s
		Bi-Att. [17]	0.2335	0.6346	0.4012	0.2998	0.7444	0.4446	15.1537	38.0932	0.158s
		TIS [27]	0.2104	0.6727	0.4623	0.2656	0.7831	0.5174	7.5218	**52.9509**	0.416s
		ViT-CX [69]	0.2427	0.6282	0.3856	0.3082	0.7332	0.4250	7.6357	52.3410	0.763s
		MDA [66]	0.2417	0.6502	0.4086	0.3082	0.7648	0.4566	21.1588	30.3994	1.803s
		LeGrad [10]	0.2401	0.6230	0.3829	0.3056	0.7292	0.4236	11.0659	43.2291	**0.005s**
		AHA (Ours)	0.1820	0.6467	0.4647	0.2370	0.7571	0.5201	9.0191	43.6451	0.016s
		AHA $T = 3$ (Ours)	**0.1514**	**0.6985**	**0.5471**	**0.1909**	**0.8138**	**0.6229**	**5.8215**	51.7770	0.212s
	Ground Truth Class	Grad-CAM [55]	0.2640	0.5574	0.2934	0.3179	0.6346	0.3168	60.3271	14.0077	0.010s
		IG [64]	0.2218	0.5783	0.3565	0.2739	0.6520	0.3781	10.5499	46.6731	0.157s
		Trans-Att. [71]	0.2338	0.5742	0.3404	0.2863	0.6482	0.3619	12.8595	43.8211	0.157s
		Bi-Att. [17]	0.2184	0.5868	0.3684	0.2714	0.6601	0.3887	13.5802	42.6191	0.158s
		TIS [27]	0.1965	0.6268	0.4303	0.2403	0.7090	0.4687	6.2001	**57.4189**	0.416s
		ViT-CX [69]	0.2363	0.5694	0.3332	0.2931	0.6361	0.3431	12.2770	48.3110	0.763s
		MDA [66]	0.2291	0.5937	0.3646	0.2834	0.6663	0.3829	21.1692	32.6573	1.803s
		LeGrad [10]	0.2240	0.5778	0.3539	0.2754	0.6519	0.3766	9.9322	47.3591	**0.005s**
		AHA (Ours)	0.1690	0.5981	0.4291	0.2117	0.6693	0.4576	8.1351	47.5630	0.016s
		AHA $T = 3$ (Ours)	**0.1410**	**0.6462**	**0.5052**	**0.1707**	**0.7245**	**0.5538**	**5.2816**	55.0229	0.212s

Table 12: Quantitative comparison of attribution methods on ImageNet using a ConvNeXt Small classifier [42] for all reference image modes $\mathbf{I}_0$ (black image, mean-colored image, blurred original image). Top-3 results per metric are highlighted (dark to light).

		Method	Deletion↓	Insertion↑	Ins. - Del.↑	Positive↓	Negative↑	Neg. - Pos.↑	ADP↓	PIC↑	Runtime
$\mathbf{I}_0 \equiv$ Black Image	Pred. Class	ScoreCAM [68]	0.3405	0.5154	0.1750	0.5082	0.7185	0.2103	56.3717	6.7919	0.821s
		AblationCAM [23]	0.2930	0.5587	0.2657	0.4337	0.7617	0.3279	51.8343	3.8980	0.742s
		GradCAM++ [13]	0.2832	0.5677	0.2845	0.4419	0.8028	0.3608	55.9903	3.7580	0.019s
		HiResCAM [26]	0.2374	0.5948	0.3574	0.3498	0.8067	0.4570	45.7085	4.7639	0.019s
		GradCAM [55]	0.2374	**0.5948**	0.3574	0.3498	**0.8068**	0.4570	45.7078	4.7600	**0.004s**
		AHA (Ours)	0.1316	0.5567	0.4251	0.1669	0.6937	0.5268	24.2854	6.6999	0.005s
		AHA $T = 3$ (Ours)	**0.1220**	0.5684	**0.4464**	**0.1536**	0.7101	**0.5565**	**22.6663**	**8.0138**	0.313s
	GT Class	ScoreCAM [68]	0.3173	0.4748	0.1575	0.4620	0.6426	0.1806	53.2696	11.7578	0.821s
		AblationCAM [23]	0.2738	0.5153	0.2415	0.3966	0.6801	0.2834	47.3906	10.1740	0.742s
		GradCAM++ [13]	0.2654	0.5229	0.2575	0.4064	0.7144	0.3081	50.3194	11.0560	0.019s
		HiResCAM [26]	0.2229	0.5479	0.3250	0.3226	0.7192	0.3966	41.0125	12.1798	0.019s
		GradCAM [55]	0.2229	**0.5479**	0.3250	0.3226	**0.7192**	0.3966	41.0117	12.1780	**0.004s**
		AHA (Ours)	0.1226	0.5173	0.3947	0.1508	0.6271	0.4762	21.8415	13.0457	0.005s
		AHA $T = 3$ (Ours)	**0.1138**	0.5275	**0.4136**	**0.1387**	0.6418	**0.5031**	**20.3470**	**14.7177**	0.313s
$\mathbf{I}_0 \equiv$ Mean-colored Image	Pred. Class	ScoreCAM [68]	0.4095	0.5748	0.1653	0.5629	0.7603	0.1974	55.5316	**6.0919**	0.821s
		AblationCAM [23]	0.3698	0.6255	0.2557	0.5129	0.8135	0.3005	48.9017	4.0800	0.742s
		GradCAM++ [13]	0.3447	0.6291	0.2844	0.4917	0.8454	0.3537	50.7066	4.6420	0.019s
		HiResCAM [26]	0.3076	0.6612	0.3536	0.4273	**0.8547**	0.4274	41.1248	5.8179	0.019s
		GradCAM [55]	0.3076	**0.6612**	0.3536	0.4273	0.8547	0.4274	41.1240	5.8180	**0.004s**
		AHA (Ours)	0.1944	0.6060	0.4116	0.2728	0.7918	0.5191	25.5903	4.8899	0.005s
		AHA $T = 3$ (Ours)	**0.1739**	0.6237	**0.4498**	**0.2406**	0.8167	**0.5761**	**24.0040**	5.9599	0.313s
	GT Class	ScoreCAM [68]	0.3805	0.5287	0.1483	0.5087	0.6779	0.1692	52.3133	11.2478	0.821s
		AblationCAM [23]	0.3439	0.5750	0.2310	0.4658	0.7218	0.2560	44.4529	10.7000	0.742s
		GradCAM++ [13]	0.3221	0.5780	0.2559	0.4502	0.7499	0.2997	45.4385	12.1920	0.019s
		HiResCAM [26]	0.2880	0.6073	0.3193	0.3922	0.7592	0.3670	36.7743	13.6277	0.019s
		GradCAM [55]	0.2880	**0.6073**	0.3193	0.3922	**0.7592**	0.3670	36.7733	**13.6280**	**0.004s**
		AHA (Ours)	0.1810	0.5591	0.3781	0.2477	0.7077	0.4600	22.9150	11.7798	0.005s
		AHA $T = 3$ (Ours)	**0.1622**	0.5743	**0.4121**	**0.2188**	0.7293	**0.5105**	**21.4147**	13.1257	0.313s
$\mathbf{I}_0 \equiv$ Blurred Original Image	Pred. Class	ScoreCAM [68]	0.3836	0.5372	0.1536	0.5333	0.7314	0.1981	56.1903	4.8119	0.821s
		AblationCAM [23]	0.3393	0.5830	0.2437	0.4792	0.7878	0.3087	46.1059	4.3200	0.742s
		GradCAM++ [13]	0.3199	0.6054	0.2855	0.4568	0.8212	0.3644	46.1080	4.7460	0.019s
		HiResCAM [26]	0.2793	0.6242	0.3449	0.3954	**0.8327**	0.4373	37.7086	5.7599	0.019s
		GradCAM [55]	0.2793	0.6242	0.3449	0.3954	0.8327	0.4374	37.7080	5.7580	**0.004s**
		AHA (Ours)	0.1963	0.6105	0.4142	0.2828	0.7993	0.5164	21.1815	6.4679	0.005s
		AHA $T = 3$ (Ours)	**0.1767**	**0.6315**	**0.4547**	**0.2507**	0.8257	**0.5749**	**19.7132**	**7.5538**	0.313s
	GT Class	ScoreCAM [68]	0.3582	0.4968	0.1386	0.4860	0.6580	0.1719	52.6074	10.1738	0.821s
		AblationCAM [23]	0.3173	0.5386	0.2213	0.4383	0.7052	0.2669	41.7112	11.0820	0.742s
		GradCAM++ [13]	0.3003	0.5589	0.2585	0.4208	0.7351	0.3143	40.9487	12.4740	0.019s
		HiResCAM [26]	0.2627	0.5761	0.3134	0.3651	0.7460	0.3809	33.4107	13.6357	0.019s
		GradCAM [55]	0.2627	0.5761	0.3134	0.3651	**0.7460**	0.3809	33.4100	13.6340	**0.004s**
		AHA (Ours)	0.1836	0.5636	0.3801	0.2587	0.7136	0.4549	18.9536	13.1017	0.005s
		AHA $T = 3$ (Ours)	**0.1655**	**0.5821**	**0.4167**	**0.2295**	0.7372	**0.5078**	**17.6002**	**14.3477**	0.313s

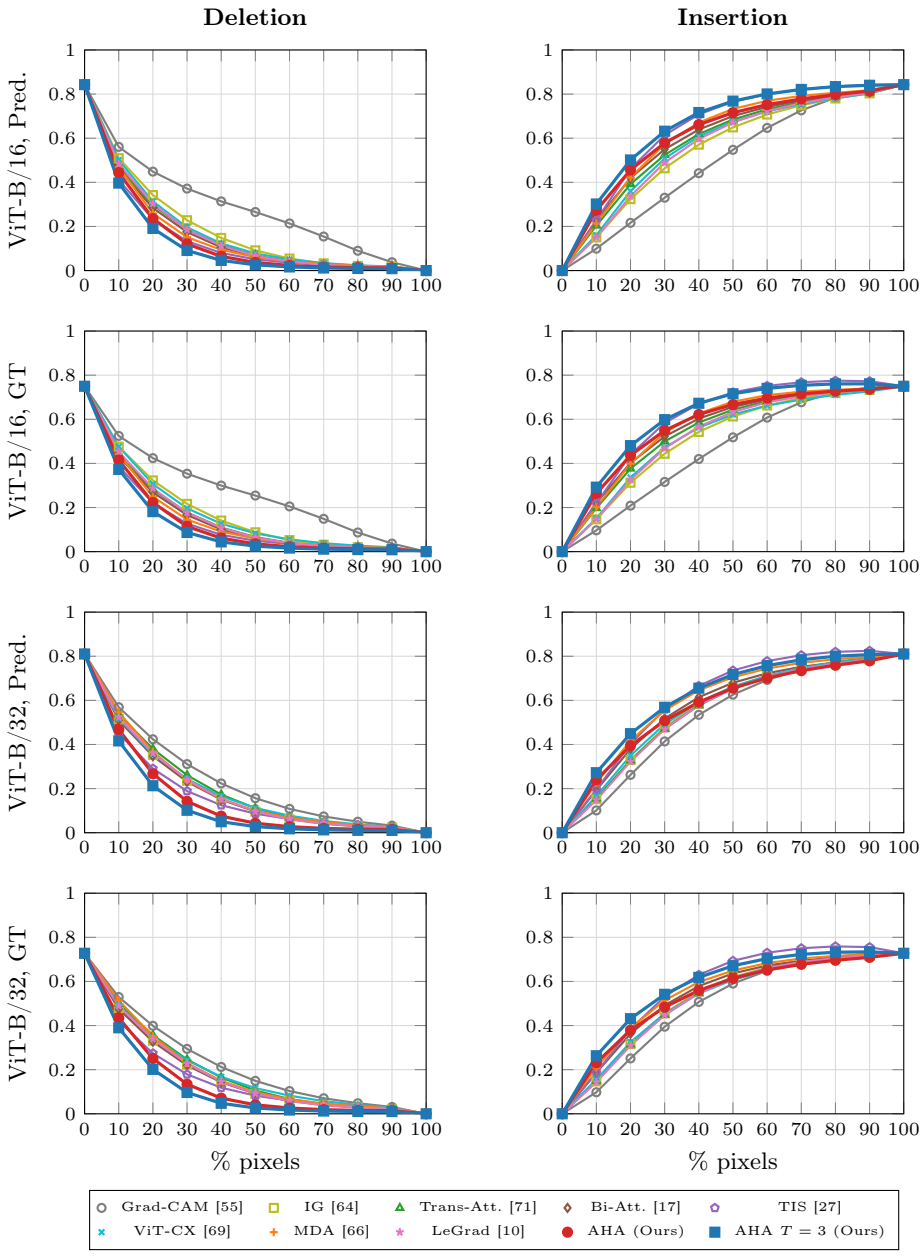


Fig. 6: Deletion and Insertion AUC curves averaged over all reference images ($\mathbf{I}_0 \in$ black, mean, blur) on ImageNet validation samples. Each row corresponds to a classifier–class combination (ViT-B/16 and ViT-B/32, predicted and ground-truth class). For Deletion (*left*), a steeper drop indicates better attribution; for Insertion (*right*), a steeper rise indicates better attribution.

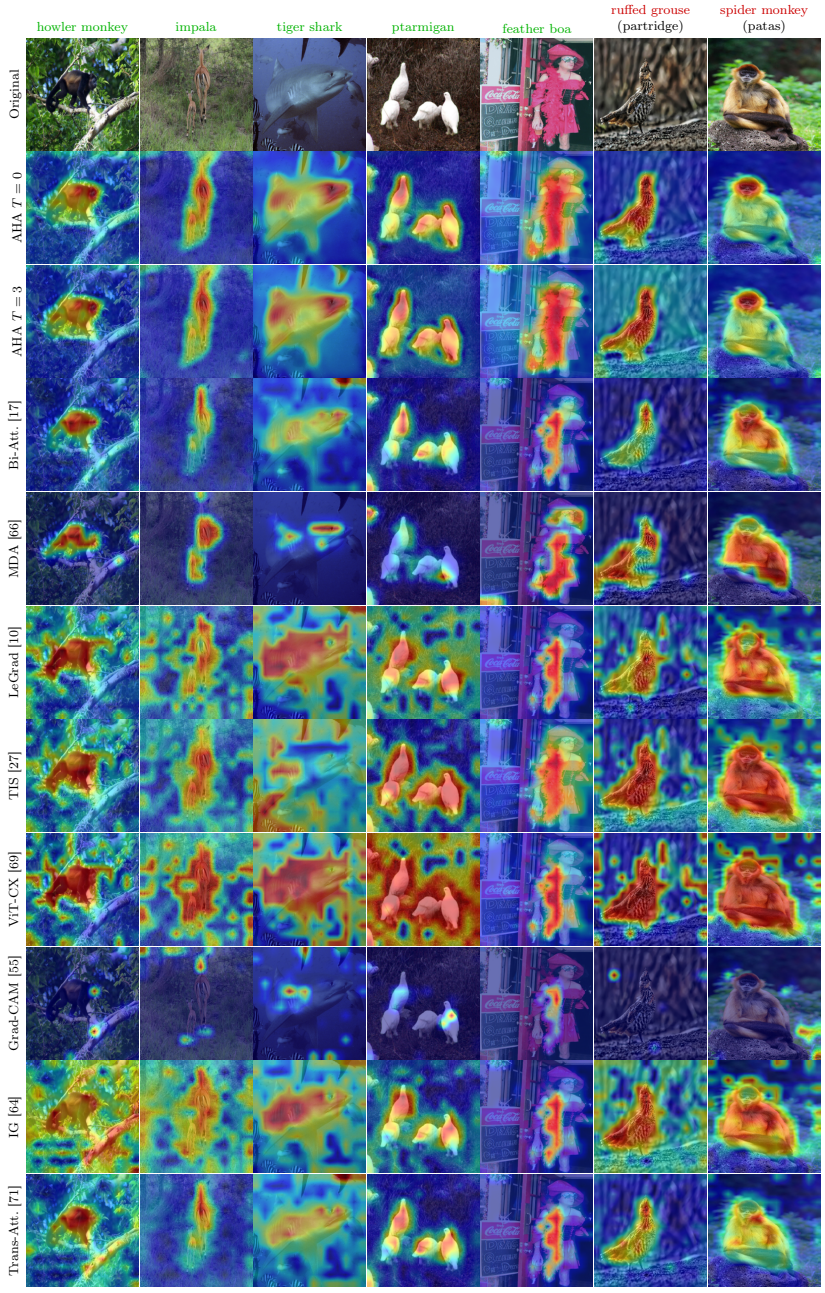


Fig. 7: Qualitative comparison of attribution maps for a ViT-B/16 classifier [25] on ImageNet [22] samples. The green / red labels mark correct / incorrect predictions (ground truth in parentheses).

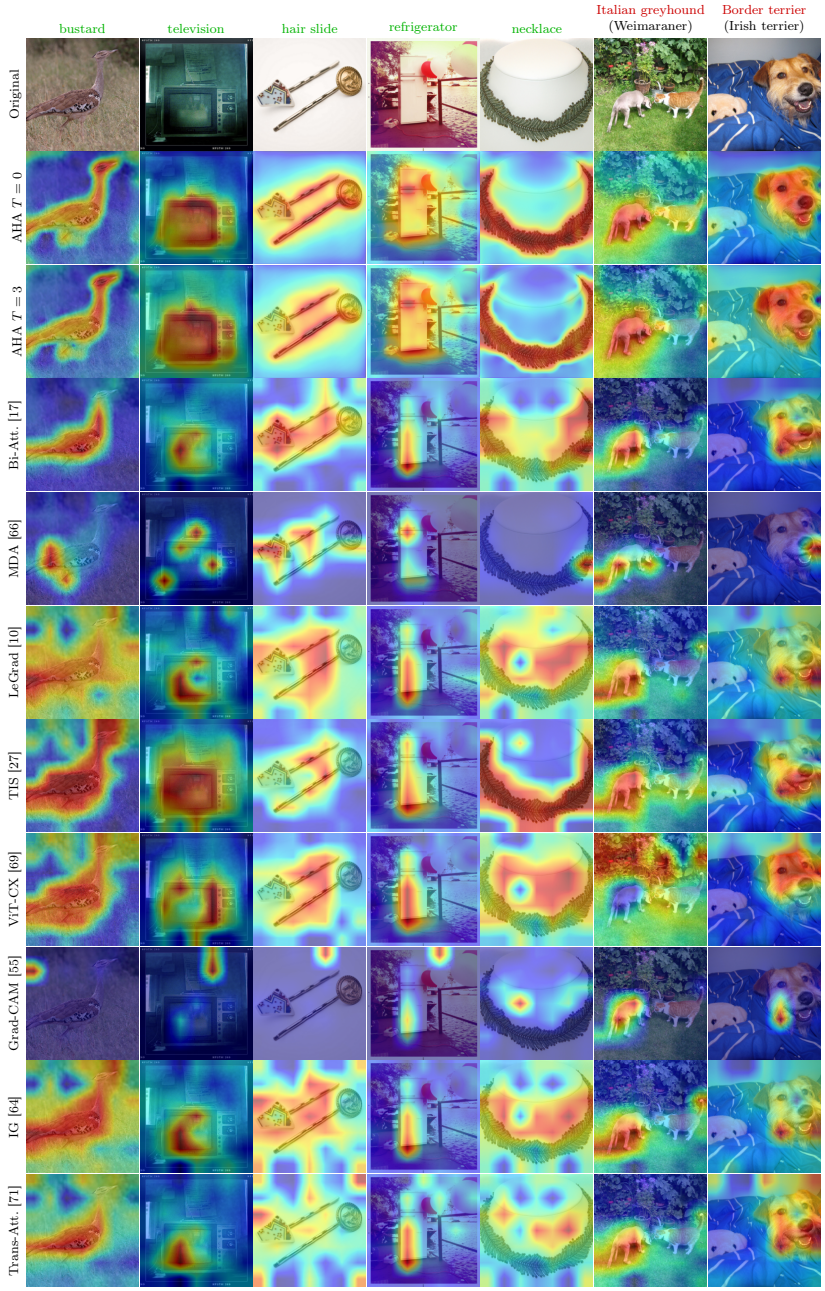


Fig. 8: Qualitative comparison of attribution maps for a ViT-B/32 classifier [25] on ImageNet [22] samples. The green / red labels mark correct / incorrect predictions (ground truth in parentheses).

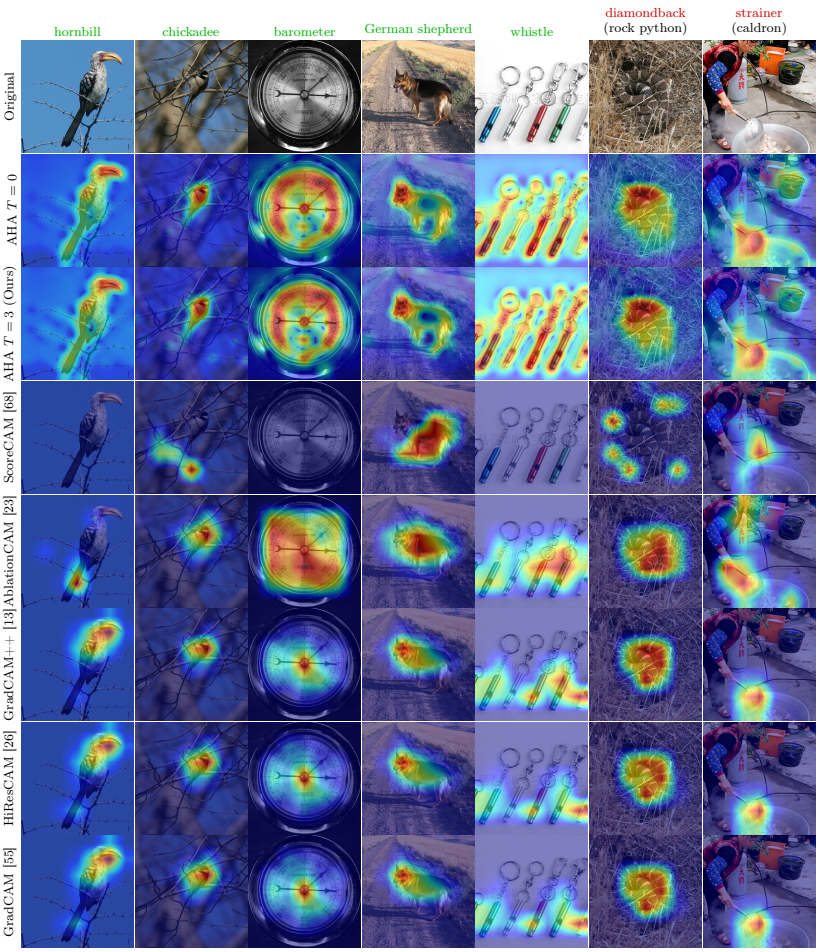


Fig. 9: Qualitative comparison of attribution maps for a ConvNeXt Small classifier [42] on ImageNet [22] samples. The green / red labels mark correct / incorrect predictions (ground truth in parentheses).