AIM: Amending Inherent Interpretability via Self-Supervised Masking

Eyad Alshami[1,2] Shashank Agnihotri[3] Bernt Schiele[1,2] Margret Keuper[1,3]
[1]Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany
[2]RTG Neuroexplicit Models of Language, Vision, and Action, Saarbrücken, Germany
[3]Data and Web Science Group, University of Mannheim, Germany
{ealshami,schiele,keuper}@mpi-inf.mpg.de, shashank.agnihotri@uni-mannheim.de

Abstract

It has been observed that deep neural networks (DNNs) often use both genuine as well as spurious features. In this work, we propose "Amending Inherent Interpretability via Self-Supervised Masking" (AIM), a simple yet interestingly effective method that promotes the network's utilization of genuine features over spurious alternatives without requiring additional annotations. In particular, AIM uses features at multiple encoding stages to guide a self-supervised, sample-specific feature-masking process. As a result, AIM enables the training of well-performing and inherently interpretable models that faithfully summarize the decision process. We validate AIM across a diverse range of challenging datasets that test both out-of-distribution generalization and fine-grained visual understanding. These include general-purpose classification benchmarks such as ImageNet100, HardImageNet, and ImageWoof, as well as fine-grained classification datasets such as Waterbirds, TravelingBirds, and CUB-200. AIM demonstrates significant dual benefits: interpretability improvements, as measured by the Energy Pointing Game (EPG) score, and accuracy gains over strong baselines. These consistent gains across domains and architectures provide compelling evidence that AIM promotes the use of genuine and meaningful features that directly contribute to improved generalization and human-aligned interpretability.

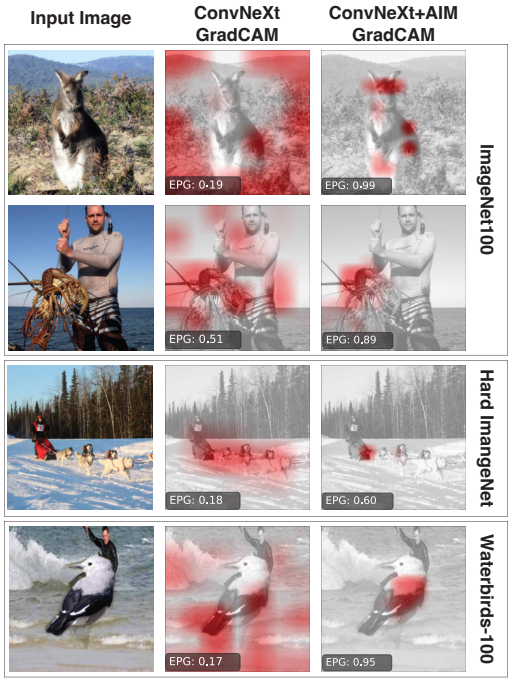

Figure 1. **AIM uses self-supervised masking for precise object localization relying only on the image label.** As shown, it outperforms baseline methods, even in challenging scenarios like the WaterBirds dataset.

1. Introduction

Modern deep neural networks (DNNs) have achieved remarkable success across domains such as Natural Language Processing and Computer Vision. Despite their impressive performance metrics, these models often use spurious features that happen to correlate with target labels in training data but lack causal relevance to the task. This phenomenon, sometimes called 'Clever Hans' behavior [20, 35]. A classic example is classification models trained to distinguish between 'land birds' and 'water birds' for the WaterBirds dataset [28, 33] that often learn to classify the background environment rather than the birds themselves, resulting in poor generalization when birds appear in atypical habitats. It is therefore desired that models leverage *genuine* features, that are distinctive, class specific, and are localized on the object. Recent research has shown that, while DNNs exhibit dependence on spurious features, they simultaneously acquire some genuine features [18]. This key insight suggests an opportunity: How can we promote the models' utilization of genuine features while suppressing spurious ones? Prior works proposed using extra annotations in the form of bound-

ing boxes, segmentation masks, or other guiding mechanisms [9, 11, 12, 22, 26, 28, 30, 34, 38, 43, 46, 53]. These mechanisms help focus the model on genuine features while ignoring spurious ones during training. However, getting these extra annotations is often nonviable.

Thus, we propose AIM (Amending Inherent Interpretability via Self-Supervised Masking), a method that encourages the model to focus on genuine features and ignore spurious ones without needing annotations beyond image labels. AIM employs a self-supervised masking mechanism that systematically identifies and prioritizes dependable feature maps of convolutional neural networks by masking out spurious features and retaining only dependable ones. Unlike previous approaches that rely on external attribution methods or require expensive additional annotations, AIM operates by applying learnable binary masks to feature maps, allowing the model itself to determine which regions to retain or discard based on task performance. Our conjecture is that, when forced to select a subset of spatial features prior to making a classification decision, a model will consider those features most dependable that generalize best, i.e., that are genuine. We confirm this hypothesis using various analyses, including Energy Pointing Game (EPG) scores and evaluations using challenging datasets that provide many spurious cues.

The AIM mechanism involves both a bottom-up processing of visual information through convolutional layers and a top-down pathway that refines feature selection. This feature refinement progressively identifies and filters out spurious features while preserving dependable ones. Importantly, this masking mechanism makes the model's decision process transparent: what is visible in the final feature representations directly causes the classification outcome, creating inherently interpretable models rather than relying on post-hoc interpretability methods.

We evaluate AIM on challenging datasets specifically designed to test models' resilience to spurious features, including Waterbirds and TravelingBirds [19]. These datasets present scenarios where background features strongly correlate with class labels during training but not during testing, challenging the models' out-of-distribution (OOD) generalization capabilities. We also validate our approach on standard fine-grained classification benchmarks such as CUB-200 [50]. Across these evaluations, AIM demonstrates significant improvements in localization accuracy (measured by the Energy Pointing Game score). It also improves classification performance in OOD scenarios, showcasing improved generalization through the use of genuine features. Our results demonstrate that by encouraging the model to narrow its selection of spatial features for classification, it improves its focus on dependable ones. AIM achieves this through a masking mechanism that produces inherently interpretable models without compromising task performance. The spatial masks learned by AIM provide clear visual evidence of the features driving the model's decisions, establishing a "what you see causes what you get" relationship between the used features and predictions.

The primary contributions of this work are threefold:

- We propose a simple yet effective self-supervised masking mechanism that guides DNNs to utilize dependable features over spurious alternatives, yielding inherently interpretable decisions while requiring only image labels.
- We demonstrate that our approach significantly improves the models' ability to localize genuine features, as quantified by Energy Pointing Game scores across multiple datasets.
- We show through extensive experiments that AIM yields improvements in challenging out-of-distribution generalization scenarios where spurious features typically cause models to fail.

2. Related Work

The prevalence of spurious correlations in DNNs, coupled with their increasing deployment in critical applications, has prompted extensive research in interpretability.

Model Guidance and Attribution Methods. Attribution methods generate attention maps [3, 4, 17, 29, 36, 39, 41, 51] that highlight important input regions contributing to the final decision, aiding in the identification of erroneous reasoning by the model. Model guidance builds on these methods to align vision systems with ground truth guidance sources [11, 12, 31, 38, 44, 45], ensuring models are 'right for the right reasons' [31]. This strategy relies on extra annotations, such as bounding boxes or attention maps [9, 11, 12, 22, 28, 30, 43, 53], which can be expensive and imperfect [11]. Several methods aim to reduce the dependency on extra annotations. For instance, cost-effective model guidance can be achieved using only a small fraction of annotated images [30]. When no extra annotations are available, approaches like [31] iteratively generate models with different reasoning but still require expert selection. Others improve explanations annotation-free simply by tuning the classification head's loss function (e.g., using binary cross-entropy) [10]. Alternatively, [2] fine-tunes the model by masking discriminative features identified by the trained model. Recent work [18] observed that DNNs, while relying on spurious correlations, still learn genuine features. However, their method assumes prior knowledge of the spurious correlation.

Content-Based Conditional Operations. Methods in this domain [14, 42, 49] constrain the model to prioritize relevant spatial regions within the input features without requiring additional annotations. Some apply masking in the feature maps during the forward pass [14, 49], pushing learned features to focus on 'regions of interest'. Others apply

masking in the input image domain [42]. For example, [14] uses mask estimators with the Gumbel-softmax trick to predict binary masks that identify crucial areas and preserve them in higher resolutions. Similarly, [49] employs mask estimators with the Gumbel-softmax trick to select and process only important spatial regions, accelerating inference. Both works [14, 49] achieve spatial selection by progressively applying the masking strategy as the input moves through the network, from the initial layers all the way to the final layers, in a bottom-up fashion. A common problem reported by both works is that the generated masks tend to be fully active, requiring additional loss functions to push them to be sparse. We hypothesize that the issue of fully active masks stems from the bottom-up approach itself. In contrast, AIM allows the network to reassess the generated feature maps across the entire architecture, utilizing the top-down approach. This naturally produces sparse masks and enables the model to use spatially sparse feature maps, enhancing explainability.

3. Proposed AIM Method

Our method comprises two main pathways: a convolutional neural network (CNN) for the bottom-up pathway with multiple encoding stages, and a top-down pathway with corresponding self-supervised masking modules. Our work proposes a novel top-down pathway that helps the CNN focus on genuine features and ignore the spurious ones learned by the encoding stages of the bottom-up pathway. It achieves this using two main components: first, a mask estimator that sparsifies the feature maps from the encoding stages, and second, a pathway that combines the sparse feature maps from different encoding stages. The following describes these main components of our method in detail. Please note, we address the task of image classification involving C classes, given a dataset $\{(x_i, y_i)\}_{i=1}^{n}$ of size n, where $x_i \in \mathbb{R}^{h \times w \times 3}$ represents input images and y_i their corresponding labels. Our approach does not require any additional annotations.

Overall Architecture. Our architecture builds on the Feature Pyramid Network (FPN) framework [21], adopting its top-down pathway structure. As illustrated in Figure 2, the model operates through two distinct pathways: a bottom-up pathway for hierarchical feature extraction and a top-down pathway for multi-scale feature integration. The bottom-up pathway employs a backbone network to generate hierarchical feature representations from input images. The top-down pathway iteratively combines these multi-scale features, propagating semantic information from the final high-level feature maps backward to earlier stages. Unlike the original FPN, which extends the top-down pathway to the highest-resolution initial feature map, we introduce a hyperparameter to control the termination depth of this pathway. This modification enables systematic analysis of how varying degrees of semantic detail from intermediate layers affect both the guidance mechanism and overall network performance. For example in our baseline ConvNeXt-Tiny backbone [24], comprising four convolutional stages $\{S_0, S_1, S_2, S_3\}$ (where S_3 marks the final stage) with spatial resolutions $\{56^2, 28^2, 14^2, 7^2\}$, the top-down pathway stages $\{T_0, T_1, T_2, T_3\}$ mirror these resolutions. We study how integrating intermediate top-down features (*e.g.* T_1, T_2) to the final stage T_3 enhances the model's self-guiding capability. In the top-down pathway, shown in Figure 2, each stage receives the output of the corresponding stage in the bottom-up pathway, processes it, and prepares it for integration with outputs from the subsequent stage in the top-down sequence. This involves passing the feature maps through: 1) a 1×1 convolutional layer to align channel dimensions across stages, 2) a 3×3 convolutional layer for spatial refinement and 4) Layer Normalization and GELU activation. The computational overhead and the increase in the number of parameters are moderate, as summarized in Appendix A.2 in the appendix.

Mask Estimation. To enable spatial selection of dependable feature regions, we incorporate a learnable mask estimator at every stage of the top-down pathway. Each estimator consists of a lightweight convolutional neural network (CNN) that predicts a binary mask using the Gumbel-Softmax trick [49]. While prior work by Verelst and Tuytelaars [49] and Hesse et al. [14] employs a bottom-up masking strategy (where binary masks iteratively select spatial regions at each stage) we adapt this concept to our top-down framework. Our empirical results demonstrate that this adaptation inherently produces spatially sparse and focused masks, enabling the network to prioritize salient regions without an additional supervision signal. As shown in Figure 2, the architecture of the mask estimators used in our method begins with a 3×3 convolutional layer, followed by three residual blocks that each utilize 3×3 convolutional layers. The output is then split into two branches: an identity branch and a global average pooling operation to capture global context information. The global context vector is expanded and concatenated with the output of the identity branch. Finally, a 1×1 convolutional layer is applied to generate the final single-channel feature maps. This single layer is passed through the Gumbel-softmax module adapted from [49] to generate the final binary mask. Each mask estimator uses the output from the corresponding stage of the backbone network as its input and generates a binary mask B with the same spatial resolution as the input feature maps. These masks highlight model-regarded dependable features in the feature maps generated by the corresponding convolutional stage. Formally, at each stage ℓ in the architecture, the bottom-up stage S_ℓ processes its input x_ℓ, producing feature maps $S_\ell(x_\ell)$. These feature

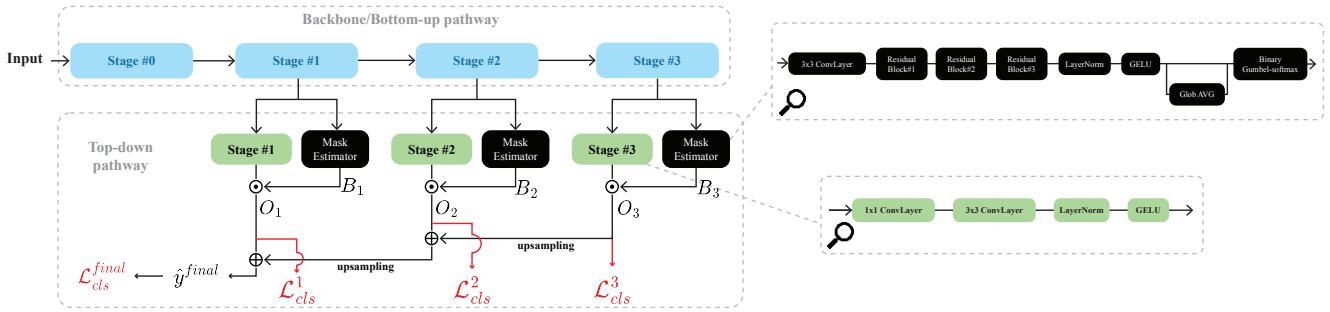

Figure 2. **Abstract Diagram of the [backbone]+AIM Architecture**. The architecture consists of a **bottom-up** backbone and a **top-down** masking pathway. The bottom-up pathway has four encoding stages ($L = 0$ to $L = 3$). The top-down pathway mirrors this structure, with each stage T corresponding to a bottom-up stage L. Each top-down stage has two parallel branches: one estimates a binary mask via a convolutional network with Gumbel-softmax, and the other processes features using a structure inspired by [21]. The estimated binary mask is element-wise multiplied with the processed features to create a spatially sparse feature map. These sparse maps are then iteratively combined with the output of the subsequent top-down stage through element-wise summation.

maps are then passed through the two branches at the corresponding top-down stage. The first branch, denoted as T_ℓ, is responsible for unifying the number of channels and post-processing the feature maps to prepare them for merging. It takes $S_\ell(x_\ell)$ as input and produces the transformed feature maps $T_\ell(S_\ell(x_\ell))$. The second branch is responsible for generating the binary mask, which highlights dependable features identified by the model. This process involves two steps. First, a soft attention decision map $A_\ell \in \mathbb{R}^{w_\ell \times h_\ell}$ is computed by a simple mask estimating module M:

$$A_\ell = M(S_\ell(x_\ell)) \tag{1}$$

Next, and following [14], to obtain the binary mask, a binary Gumbel-softmax module G is applied element-wise to A_ℓ, resulting in $B_\ell \in \{0,1\}^{w_\ell \times h_\ell}$:

$$B_\ell = G(A_\ell) \tag{2}$$

Finally, the spatially sparse output of the top-down stage ℓ is then computed by element-wise multiplying the processed feature maps from the feature processing branch with the binary mask from the mask estimator:

$$O_\ell = T_\ell(S_\ell(x_\ell)) \odot B_\ell \tag{3}$$

where $\odot$ denotes element-wise multiplication. This operation results in O_ℓ, which retains only the dependable features as determined by the binary mask, effectively filtering out spatial regions with spurious features.

Top-Down Sparse Feature Fusion Along the top-down pathway, the output of each stage is up-sampled through nearest-neighbor interpolation to match the resolution of the next lower stage and then merged with its feature maps through element-wise summation. This continues down the pathway, progressively integrating multi-scale information. The final aggregated feature maps, containing reliable multi-scale features, are used for classification.

Supervising The Mask Estimators. Since our method does not rely on additional annotations, we supervise the mask estimators indirectly using the classification loss computed on the masked feature maps O_ℓ. At each stage of the top-down pathway, this loss is applied before merging with higher-stage feature maps. This strategy ensures that each stage independently identifies and learns important regions based solely on the feature maps available up to that stage. We pass these feature maps through a classifier f_ℓ to obtain the predicted class probabilities $\hat{y}^{(\ell)}$:

$$\hat{y}^{(\ell)} = f_\ell(O_\ell) \tag{4}$$

And then compute the classification loss at each stage, $\mathcal{L}_{\text{cls}}^{(\ell)}$. At the final stage, the merged sparse feature maps, combining outputs from all previous stages, are passed through the final classifier f_{final} to obtain the final predicted class probabilities $\hat{y}^{\text{final}}$:

$$\hat{y}^{\text{final}} = f_{\text{final}}(O_{\text{final}}) \tag{5}$$

During training, our model self-guides to highlight spatial regions with dependable features within each stage's output. By enabling the network to select these regions across all layers, we empirically show that this improves classification performance and enables the reliance on spatially sparse maps, leading to transparent and inherently interpretable decision-making.

Optional Mask Annealing. Our approach naturally generates sparse masks, but enforcing additional sparsity during training on challenging out-of-distribution datasets, such as Waterbirds and TravelingBirds, improved performance and produced more focused masks. This is done by applying a mean-squared loss on the number of active elements in the generated masks using a threshold τ_i as follows:

$$\mathcal{L}_{\text{masks}_i} = (r_i - \tau_i)^2 \quad \text{where} \quad r_i = \frac{\sum_{j=0,k=0}^{B_h^i, B_w^i} \mathbb{1}(B_{j,k}^i = 1)}{\sum_{j,k}^{B_h^i, B_w^i} 1} \tag{6}$$

Where r_i is the ratio of active elements in the generated binary mask B^i, and τ_i is a threshold hyperparameter that can be selected for each stage's mask estimator. We use a masking annealing technique to help the network gradually adapt to sparsity constraints without disrupting learning. Training begins with fully active masks (i.e. $\tau_i = 1.0$) and progressively lowers the active-area loss threshold each epoch until it reaches a target value (e.g. $\tau_i = 0.35$), which is then held for the remainder of training. The annealing duration is treated as a hyperparameter. This strategy improves mask quality and stabilizes learning. (For more details, see Appendix D). Based on this setup, the final loss is defined as:

$$\mathcal{L}_{Total} = \lambda \sum_{i=L}^{\ell} \mathcal{L}_{\text{masks}_i} + \sum_{i=L}^{\ell} \mathcal{L}_{\text{cls}}^{(i)} \tag{7}$$

Here, L is the index of the highest stage, and ℓ denotes the final stage we aim to reach in the top-down pathway. The parameter λ, set to 6 in all experiments, weights the mask's active-area loss to be on the same order of magnitude as the classification loss. For a full list of hyperparameters, see Appendix A.

4. Experiments

In this section we show that AIM helps to retain in-domain performance while significantly boosting out-of-domain performance. First, however, we describe some important implementation details (more details in Appendix A).

AIM Architectural Variants. As detailed in Section 3, we parameterize the top-down pathway's depth by the number of stages traversed, denoting these variants as "Backbone+AIM [*index*]", where *index* signifies the stage T where propagation ceases. For instance, with ResNet50, which has five convolutional stages, including the stem cell, we implement two main variants: ResNet50+AIM (2) incorporates feature maps from stages 4, 3, and 2, while ResNet50+AIM (3) includes only feature maps from stages 4 and 3.

Baselines. To evaluate our approach, we tested the effect of applying AIM across various backbone architectures by comparing the performance of each backbone with and without AIM integration. We utilized ConvNeXt-tiny [24], ResNet-50 [13], and ResNet-101 [13]. All of these models are pre-trained on ImageNet-1k [5].

Evaluation Metrics. To assess how effectively AIM promotes dependable feature learning, we evaluate spatial localization using the Energy Pointing Game (EPG) score [51]. This metric, based on attribution maps (e.g., GradCAM [36], Guided GradCAM [37], or other attribution methods) and ground-truth binary masks, calculates the ratio of attribution within the mask's active region to the total attribution. For implementation details, see Appendix A.5. In addition, we show that our method not only preserves but also improves classification accuracy.

Mask Annealing via Active-Area Loss. As detailed in Section 3, in addition to the stage T parameterization, we employ progressive mask sparsification via threshold annealing during training, reducing the initial 100% active area to either 35% or 25%. This introduces a new parameter, τ, representing the final mask retention. Consequently, our models are denoted as "backbone+AIM [stage T, threshold τ]". For example, ResNet50+AIM [2, 25%] signifies operation through stage $T = 2$ with 25% mask retention.

Datasets. We evaluate our method on two categories of datasets. First, to test its robustness to spurious correlations, we use the synthetic Waterbirds (95% and 100% versions) [28, 33] and Travelingbirds [19] datasets. Second, to assess the broader adaptability and effectiveness of AIM, we use standard benchmarks including ImageNet100 [47], Hard-ImageNet [27], and the fine-grained Caltech-UCSD Birds-200-2011 (CUB-200-2011) [50]. Further details are provided in Appendix A.4.

4.1. Results on Out-Of-Domain Datasets

The Waterbirds and TravelingBirds datasets contain synthetic spurious correlations, causing models to incorrectly rely on background cues rather than foreground objects. As Figure 4 illustrates, our proposed AIM mechanism consistently surpasses baseline backbone models. Vanilla backbone performance significantly degrades due to these biases; however, models integrated with AIM show notable improvements in both EPG and accuracy across all tested out-of-domain datasets. The primary motivation of AIM, detailed in Section 1, is to enhance the localization of genuine image features, thereby improving interpretability without compromising accuracy. The improved accuracy across various backbones emerges as an additional beneficial outcome. The self-supervised masking strategy employed by AIM enables models to consistently identify and rely on dependable features. Figure 3 visually demonstrates this, contrasting baseline models, which are often distracted by misleading background cues, with AIM-equipped models that reliably emphasize dependable regions. To confirm this visual improvement is perceived by humans, we conducted a user study that showed participants preferred our model's attribution maps over the baseline in 70.7% of cases ($p < 0.00001$), providing strong evidence of more human-aligned interpretability (see Appendix D.6 for details). Quantitatively, higher EPG scores, computed using dataset-provided binary masks, confirm this improved localization. We also evaluate other attribution methods and observe similar EPG improvements on the Waterbirds-95% dataset using Guided GradCAM and Guided Backprop, as shown in Appendix B.2, further confirming the robustness of our localization gains. These scores validate our hy-

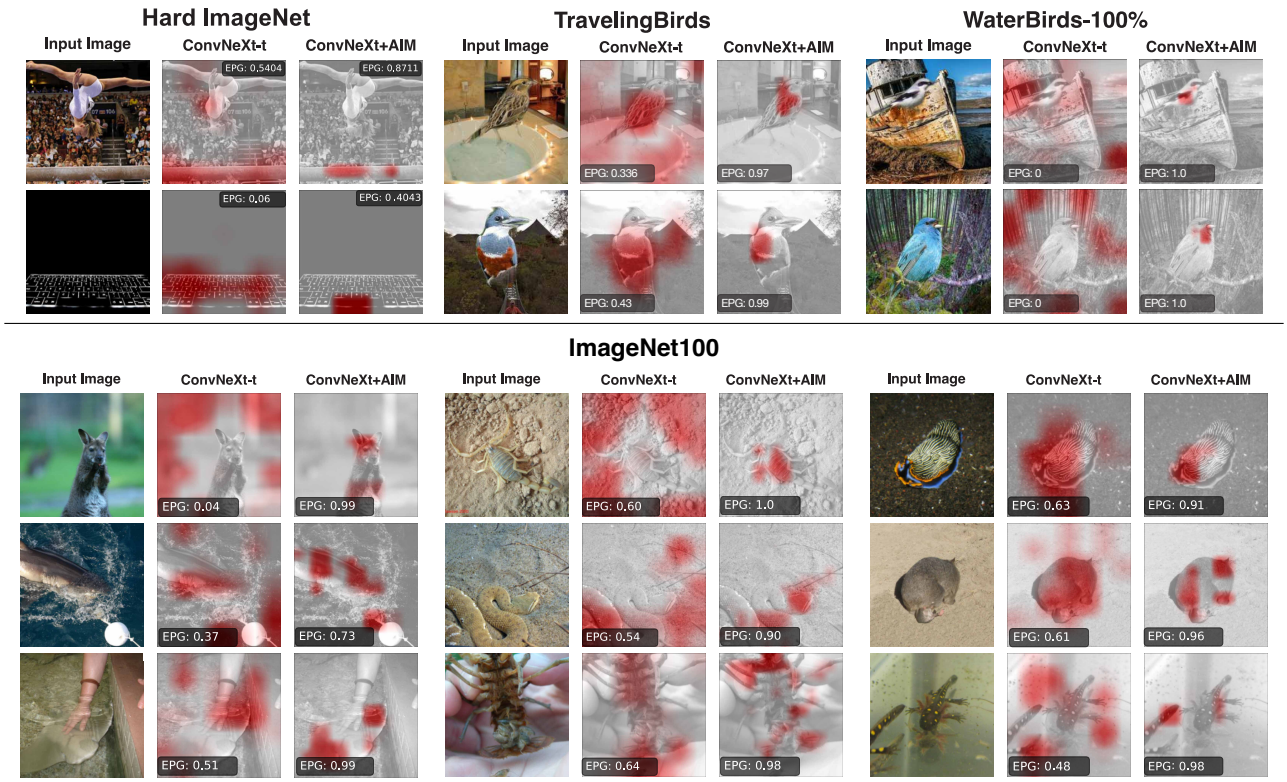

Figure 3. **Models amended with AIM consistently exhibit enhanced localization of genuine features, effectively suppressing spurious cues in both in-domain and out-of-domain scenarios.** A qualitative visualization of Grad-CAM heatmaps comparing baseline ConvNeXt-tiny models and ConvNeXt-tiny+AIM models across HardImagenet (classes shown: Balance Beam, Space Bar), TravelingBirds, WaterBirds-100%, and ImageNet100 datasets. The EPG scores, with a range of 0.0 to 1.0, are indicated on each heatmap. For more qualitative results, see Appendix C.

pothesis from Sec.1 regarding the dependability of features identified by AIM. As summarized visually in Figure 4 and detailed with precise metrics in Table 1, we see substantial EPG score improvements: approximately 6% for Waterbirds-95%, 30% for Waterbirds-100%, and 10% for TravelingBirds. Accuracy gains are equally notable, reaching around 10% for Waterbirds-95%, 40% for Waterbirds-100%, and 18% for TravelingBirds. Furthermore, Figure 5 provides a detailed per-sample analysis of EPG scores for baseline models versus models with AIM. It demonstrates that across all datasets, the majority of individual samples exhibit improved EPG scores. While overall improvements are substantial, a subset of examples maintained comparable performance, particularly when baseline EPG was already high, and a minimal number of instances showed slight EPG decreases. Additional comparisons against other relevant methods are provided in the Appendix B.4.

4.2. Comprehensive Evaluation on Diverse Classification Tasks

To evaluate the adaptability and effectiveness of our proposed AIM mechanism, we conducted experiments on a range of classification benchmarks, from fine-grained tasks to broader, general-purpose datasets.

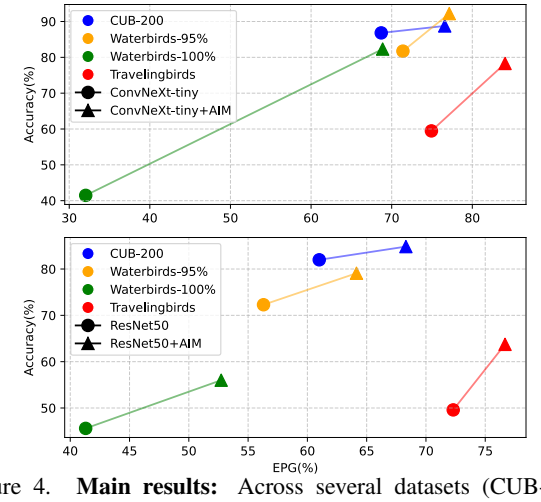

Figure 4. **Main results:** Across several datasets (CUB-200, Waterbirds-95%, Waterbirds-100%, Travelingbirds) and architectures (ConvNeXt-tiny, ResNet50) our AIM approach consistency outperforms the respective baseline in both accuracy as well as interpretability as measured by the Energy Pointing Game score. On WaterBirds, following [33], we report the worst-group accuracy.

First, we tested our method on the CUB-200 dataset, which poses a challenging fine-grained classification task

Table 1. **Average Test Accuracies for ConvNeXt+AIM Configurations.** Comparison of the ConvNeXt-tiny baseline against our AIM-enhanced models on multiple benchmarks. The method shows significant gains, especially in worst-group accuracy on Waterbirds, highlighting its effectiveness in mitigating spurious correlations. All values are mean accuracy (%) ± standard deviation.

Model	ImageNet100		Hard-ImageNet		Waterbirds				TravelingBirds	
					100%		95%			
	Acc	EPG	Acc	EPG	WG-Acc	EPG	WG-Acc	EPG	Acc	EPG
ConvNeXt-t	89.2 (±0.1)	91.4 (±0.3)	96.2 (±0.2)	36.6 (±0.5)	39.6 (±5.4)	57.2 (±6.0)	81.6 (±3.2)	68.3 (±3.2)	59.5 (±0.8)	74.4 (±0.6)
ConvNeXt-t+AIM[1, 25%]	90.5 (±2.1)	91.5 (±0.1)	97.1 (±0.3)	38.8 (±0.9)	73.6 (±4.5)	60.1 (±1.3)	91.2 (±0.8)	**77.1 (±5.2)**	77.1 (±0.3)	79.0 (±0.7)
ConvNeXt-t+AIM[1, 35%]	90.5 (±0.1)	89.1 (±0.2)	**97.3 (±0.2)**	33.2 (±0.5)	77.1 (±4.4)	57.2 (±1.3)	90.7 (±0.7)	63.0 (±1.2)	71.5 (±1.3)	72.6 (±1.5)
ConvNeXt-t+AIM[2, 25%]	90.1 (±2.3)	**92.8 (±0.8)**	96.8 (±0.5)	**40.1 (±1.5)**	74.0 (±5.0)	58.0 (±1.3)	**92.7 (±1.2)**	75.0 (±6.0)	**77.4 (±0.2)**	**85.0 (±2.0)**
ConvNeXt-t+AIM[2, 35%]	**90.7 (±0.0)**	91.8 (±0.1)	97.1 (±0.1)	33.6 (±1.1)	**78.1 (±2.3)**	**68.5 (±3.6)**	92.3 (±0.6)	71.7 (±6.4)	71.0 (±0.4)	77.7 (±0.4)

requiring models to identify subtle and localized visual features. As shown in Figure 4, incorporating our AIM mechanism improves localization performance significantly: ConvNeXt-tiny+AIM achieves an approximate 6% increase in EPG score over the baseline ConvNeXt-tiny model, while ResNet+AIM improves localization by around 9% compared to the baseline ResNet model. These localization improvements are accompanied by a slight accuracy increase of about 2–3%. For full details see Appendix B.3.

Next, to validate the broader applicability beyond the domain of bird datasets, we evaluated AIM on general-purpose image classification benchmarks. We conducted experiments using ConvNeXt-tiny on ImageNet100 [47] and the challenging HardImageNet [27]. As summarized in Table 1, our method shows consistent benefits across diverse domains. On ImageNet 100, it improves the EPG score by nearly 3 points while maintaining baseline accuracy, reinforcing our core claim of enhanced localization. On the more difficult HardImageNet, it boosts both accuracy and EPG, confirming its robustness as a domain-agnostic mechanism.

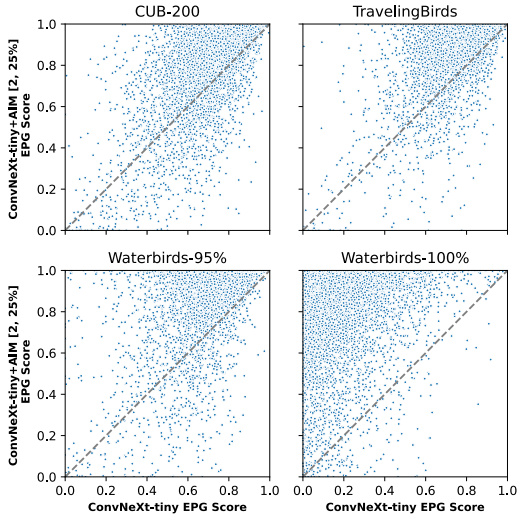

Figure 5. EPG scores per sample are plotted for baseline model (x-axis) v/s model amended with AIM (y-axis). We observe at a per-sample level for each of the four datasets that majority for the samples the EPG scores are improved by amending the model with our proposed AIM.

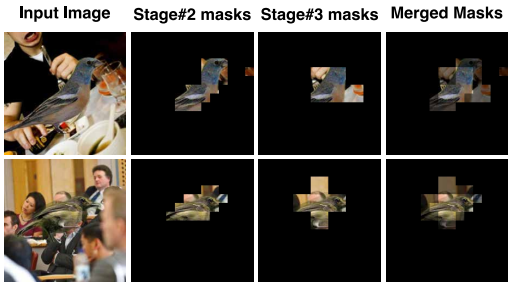

Figure 6. Illustration of masks learned at the two stages within the network (using ConvNeXt-tiny+AIM [2, 25%]), along with the final merged mask for each image. These merged masks highlight the sparse regions within the corresponding feature maps.

5. Inherent Interpretability With Self-Supervised Masking

Our AIM mechanism offers inherent interpretability, which we visualize by depicting input images alongside their corresponding masks from the top-down pathway in Figure 6. This interpretability arises from the self-supervised masking performed by the mask estimator within AIM. Furthermore, combined masks clearly illustrate how sparse feature maps from different stages of the top-down pathway are merged.

Figure 7 visualizes the evolution of these masks over epochs. The mask estimator initially starts with random values, but as training progresses, relying solely on the classification loss from image labels, it learns to focus on dependable features within the feature maps. As discussed in Section 4, these dependable features correspond to genuine features, indicated by EPG scores. Conversely, if the model learns incorrect masking, low EPG scores reflect non-genuine features and result in lower accuracy scores. Since AIM's self-supervised masking mechanism is part of the model's forward pass, visualizing these masks directly reveals the basis of the model's decisions. This establishes

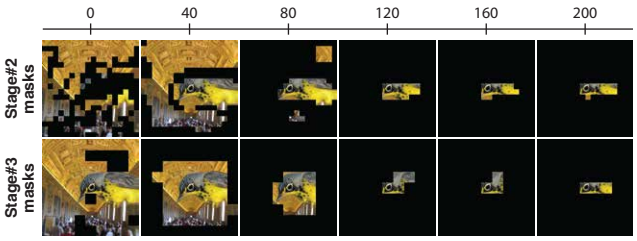

Figure 7. Visualization of the evolution of learned masks at two different stages (Stage #2 and Stage #3) of a ConvNeXt-tiny+AIM [2, 25%] model throughout the training epochs. As training progresses, the masks gradually become more sparse and accurately localized, highlighting the model's improved ability to identify and focus on regions containing genuine features in a self-supervised manner.

Table 2. **Lower performance with Bottom-up Guiding Approach.** Result of the bottom-up ConvNeXt-t+AIM applying the masking mechanism in the second and third stages in the vanilla ConvNeXt-tiny, compared to the top-down Refocs+ConvNeXt-t model.

Model	CUB-200 (%)
bottom-up masking [1, 25%] ConvNeXt-t	72.79 (±8.51)
ConvNeXt-t+AIM [1, 25%]	**88.82 (±0.213)**
bottom-up masking [2, 25%] ConvNeXt-t	84.00 (±1.38)
ConvNeXt-t+AIM [2, 25%]	**88.677 (±0.25)**

a clear "what you see causes what you get" relationship between features and predictions.

6. Analysis and Ablation

The following further explores the effectiveness of AIM.

6.1. *Top-down* approach v/s *Bottom-up* approach

Inspired by [49], we initially tested a bottom-up masking approach, utilizing the same mask estimators described in Section 3 but without a top-down pathway. In this setup, each convolutional stage of the backbone model had two branches: the original convolutional path and a mask estimator. The latter predicted a binary mask, applied to the convolutional output to create spatially sparse feature maps that proceeded to the next stage. Unlike [49], we did not employ a skip-connection to convert sparse feature maps back into dense ones, aiming to preserve inherent explainability. However, this bottom-up guiding method performed poorly on the CUB-200 dataset, as shown in Table 2, where bracketed numbers indicate the used stage and annealing. Furthermore, the generated masks tended to remain fully active despite applying the mask active-area loss (see Appendix D.4 for further analysis), unlike the naturally focused masks produced by the top-down approach.

6.2. Does AIM have a center bias?

To investigate potential center bias [8] in our experiments, we tested models on images with birds positioned at the edges rather than center-frame. Table 3 shows

Table 3. **AIM does not exploit the center-bias** AIM manages to detect and focus on the bird achieving higher results compared to the vanilla ConvNeXt-tiny model

Model	CUB-200 (%)
ConvNeXt-tiny	76.98 (±0.18)
ConvNeXt-t+AIM [1, 25%]	**79.33 (±0.45)**

that while both vanilla ConvNeXt-tiny and ConvNeXt-tiny+AIM experienced performance decreases compared to center-cropped images, AIM still outperformed the baseline by approximately 2.5%. Furthermore, as depicted in Figure 8, AIM continued to generate masks focused on birds despite partial visibility, confirming our approach does not depend on center bias for its effectiveness.

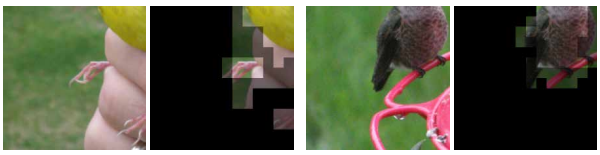

Figure 8. **AIM models do not have a center-bias.** This illustration shows the merged masks generated by ConvNeXt-t+AIM (2, 25%) on two images from CUB-200.

7. Conclusion

In this work, we propose Amending Inherent Interpretability via Self-Supervised Masking (AIM), a simple yet effective method that encourages networks to focus on dependable rather than spurious features via a self-supervised feature-masking process. Evaluated using the Energy Pointing Game (EPG) score on out-of-distribution and fine-grained classification tasks, AIM improves localization on dependable features without sacrificing accuracy or requiring annotations beyond class labels. AIM produces inherently interpretable models by integrating sparse, top-down feature selection directly into the forward pass. It consistently improves both accuracy and localization across diverse datasets and architectures, with minimal overhead. These results highlight AIM's potential as a lightweight and scalable approach to training models that are robust, generalizable, and aligned with meaningful visual cues.

Future Work. We plan to extend AIM to Vision Transformers [6] by either reshaping patch embeddings into spatial feature maps or leveraging the hierarchical structure of Swin-Transformers [23] for more seamless integration.

8. Acknowledgment.

Funded in part by the DFG (German Research Foundation, RTG 2853/1). S.A. and M.K. acknowledge support by DFG Research Unit 5336 Learning2Sense.

References

[1] Shashank Agnihotri, Shashank Priyadarshi, Hendrik Sommerhoff, Julia Grabinski, Andreas Kolb, and Margret Keuper. Roll the dice: Monte carlo downsampling as a low-cost adversarial defence, 2024. 15

[2] Saeid Asgari, Aliasghar Khani, Fereshte Khani, Ali Gholami, Linh Tran, Ali Mahdavi Amiri, and Ghassan Hamarneh. Masktune: Mitigating spurious correlations by forcing to explore. *Advances in Neural Information Processing Systems*, 35:23284–23296, 2022. 2

[3] Moritz Böhle, Mario Fritz, and Bernt Schiele. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10329–10338, 2022. 2

[4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 8

[7] Ryan Farrell. Cub-200-2011 segmentations, 2022. 14, 15

[8] Mishal Fatima, Steffen Jung, and Margret Keuper. Corner cases: How size and position of objects challenge imagenet-trained models. In *Synthetic Data for Computer Vision Workshop@ CVPR 2025*. 8

[9] Thomas Fel, Ivan F Rodriguez Rodriguez, Drew Linsley, and Thomas Serre. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in neural information processing systems*, 35:9432–9446, 2022. 2

[10] Siddhartha Gairola, Moritz Böhle, Francesco Locatello, and Bernt Schiele. How to probe: Simple yet effective techniques for improving post-hoc explanations, 2025. 2

[11] Yuyang Gao, Tong Steven Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Zhao Liang. Res: A robust framework for guiding visual explanation. In *proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 432–442, 2022. 2

[12] Yuyang Gao, Tong Steven Sun, Liang Zhao, and Sungsoo Ray Hong. Aligning eyes between humans and deep neural network through interactive attention alignment. *Proceedings of the ACM on Human-Computer Interaction*, 6 (CSCW2):1–28, 2022. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[14] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Content-adaptive downsampling in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2023. 2, 3, 4

[15] J Hoffmann, S Agnihotri, Tonmoy Saikia, and Thomas Brox. Towards improving robustness of compressed cnns. In *ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL)*, 2021. 15

[16] Jeremy Howard. Imagewoof: a subset of 10 classes from imagenet that aren't so easy to classify, 2019. 15

[17] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. 2

[18] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022. 1, 2

[19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 2, 5, 15, 31

[20] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019. 1

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 3, 4, 20

[22] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*, 2018. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 8

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 3, 5

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 13

[26] Masahiro Mitsuhara, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Embedding human knowledge into deep neural network via attention map. *arXiv preprint arXiv:1905.03540*, 2019. 2

[27] Mazda Moayeri, Sahil Singla, and Soheil Feizi. Hard imagenet: Segmentations for objects with strong spurious cues, 2022. 5, 7, 15

[28] Suzanne Petryk, Lisa Dunlap, Keyan Nasseri, Joseph Gonzalez, Trevor Darrell, and Anna Rohrbach. On guiding visual attention with language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18092–18102, 2022. 1, 2, 5, 14, 18, 32

[29] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *proceedings of the IEEE/CVF winter*

conference on applications of computer vision, pages 983–991, 2020. 2

[30] Sukrut Rao, Moritz Böhle, Amin Parchami-Araghi, and Bernt Schiele. Studying how to efficiently and effectively guide models with explanations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1922–1933, 2023. 2, 18

[31] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017. 2

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 15

[33] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 1, 5, 6, 14, 29, 30

[34] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020. 2

[35] Thomas A Sebeok and Robert Ed Rosenthal. The clever hans phenomenon: Communication with horses, whales, apes, and people. *Annals of the New York Academy of Sciences*, 1981. 1

[36] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2, 5, 15

[37] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019. 5, 15

[38] Haifeng Shen, Kewen Liao, Zhibin Liao, Job Doornberg, Maoying Qiao, Anton Van Den Hengel, and Johan W Verjans. Human-ai interactive and continuous sensemaking: A case study of image classification using scribble attention maps. In *extended abstracts of the 2021 CHI conference on human factors in computing systems*, pages 1–8, 2021. 2

[39] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017. 2

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. 15

[41] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 2

[42] Saeid Asgari Taghanaki, Aliasghar Khani, Fereshte Khani, Ali Gholami, Linh Tran, Ali Mahdavi-Amiri, and Ghassan Hamarneh. Masktune: Mitigating spurious correlations by forcing to explore, 2022. 2, 3

[43] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer, 2020. 2

[44] Stefano Teso. Toward faithful explanatory active learning with self-explainable neural nets. In *Proceedings of the Workshop on Interactive Adaptive Learning (IAL 2019)*, pages 4–16. CEUR Workshop Proceedings, 2019. 2

[45] Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 239–245, 2019. 2

[46] Stefano Teso, Öznur Alkan, Wolfgang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 6: 1066049, 2023. 2

[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 5, 7, 15

[48] Robert van der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1866–1876, 2023. 18

[49] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 2, 3, 8, 22, 34

[50] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5, 14

[51] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020. 2, 5, 15

[52] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *The Twelfth International Conference on Learning Representations*, 2024. 18

[53] Ziyan Yang, Kushal Kafle, Franck Dernoncourt, and Vicente Ordonez. Improving visual grounding by encouraging consistent gradient-based explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19165–19174, 2023. 2

[54] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 14, 15