Scale Matters: Attribution Meets the Wavelet Domain to Explain Model Sensitivity to Image Corruptions

Gabriel Kasmi, Laurent Dubus, Yves-Marie Saint Drenan, Philippe Blanc

Scale Matters: Attribution Meets the Wavelet Domain to Explain Model Sensitivity to Image Corruptions

Gabriel Kasmi[1,2] **Laurent Dubus**[2,3] **Yves-Marie Saint Drenan**[1] **Philippe Blanc**[1]
[1]MINES Paris, Université PSL Centre Observation Impacts Energie (O.I.E.)
[2]RTE France
[3]WEMC (World Energy & Meteorology Council, UK)
[1]{firstname.lastname}@minesparis.psl.eu
[2]{firstname.lastname}@rte-france.com

Abstract

Neural networks have shown remarkable performance in computer vision applications, but their deployment in real-world scenarios is challenging due to their sensitivity to image corruptions. Existing attribution methods are uninformative for explaining the sensitivity to image corruptions, while the literature on robustness only provides model-based explanations. However, the ability to scrutinize models' behavior under image corruptions is crucial to increase the user's trust. Towards this end, we introduce the **W**avelet s**C**ale **A**ttribution **M**ethod (**WCAM**), a generalization of attribution from the pixel domain to the space-scale domain. Attribution in the space-scale domain reveals where *and* on what scales the model focuses. We show that the WCAM explains models' failures under image corruptions, identifies sufficient information for prediction, and explain how zoom-in increases accuracy.

1 Introduction

Deep neural networks have become the *de facto* standard for numerous computer vision applications. However, there is a growing consensus that these models cannot be safely deployed in real-world applications [9]. It is partly because deep learning systems are sensitive to small image corruptions, *i.e.*, blur, pixelation, compression [25, 18], thus hindering safe generalization to unseen data. In this context, increasing the trust necessitates being able to analyze the model's decision at the instance-based label (*i.e.*, for one prediction) [50]. To achieve this, we need to know where *and* what models see, especially if what they see makes their decision process unreliable [39]. Therefore, an explainability method, informative when image corruptions disrupt the model's decision process, is necessary to improve the trustworthiness of deep learning systems.

On the one hand, explainable AI (XAI) seeks to explain a model's decision at the instance-based level. Attribution methods [5], which consist in identifying the most important features in the input, have improved the understanding of the decision process of deep learning models. On the other hand, a vast body of literature highlighted the higher tendency of non-robust models to rely on high-frequency components [67, 63, 6, 71].

As illustrated in Figure 1, existing explainability approaches are uninformative for explaining model sensitivity to image corruptions, as attribution methods usually identify *where* on the image domain models focus. On the other hand, robustness studies are limited to explaining robustness at the model-based level (*i.e.*, highlighting the average behavior of a model on a dataset). Making explanations informative under image corruptions requires showing what models see by bridging the gap between model-based robustness assessment and instance-based explainability.

Preprint. Under review.

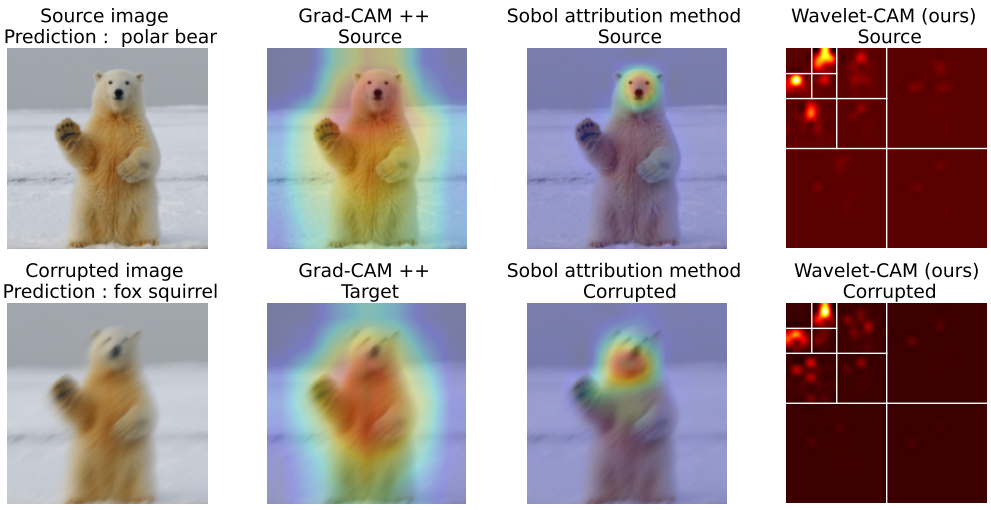


Figure 1: State-of-the art explanations (Grad Cam ++ [52] and Sobol [12], middle columns) are uninformative to explain why the motion blur mislead the model (leftmost column). Our WCAM (rightmost column) shows that due to the blurring, the model relied more on details at the 2-4 pixel scale on the body and the paw and less on the shapes on the head (4-8 pixel scale). The blurring led the model to pay less attention to shape and more to texture, where it misled a bear's fur with a squirrel's fur. Expanding attribution to the scale-space domain helps to explain *why* corruptions affect predictions.

Wavelets decompose images in terms of *scales*, which enables highlighting their structural properties [44]. By introducing the **W**avelet s**C**ale **A**ttribution **M**ethod (**WCAM**), we highlight simultaneously *what* are the important scales and *where* they are located. The WCAM builds on the Sobol attribution method introduced by [12]. We quantify the importance of the image's wavelet coefficients in a model's decision by estimating the total Sobol indices corresponding to these coefficients. To estimate the total Sobol indices, we randomly perturb the wavelet transform of the input image, thus expanding attribution from the pixel (image) domain to the space-scale (wavelet) domain.

We show that the WCAM answers *why models fail to classify an image under image corruptions correctly*. In particular, we document that models focus on a limited set of coefficients in the space-scale domain. A distribution shift typically induces a disruption in the important *scales*, not the localization of the important components. This behavior is consistent across various model architectures and training methods (e.g., adversarial [42, 64, 53] and robust [26, 28, 20] approaches). Additionally, we show that we can use the WCAM to identify the location (in space and scale) of the sufficient information to make a correct prediction. This further helps understand what is meaningful in the input and could have applications for transfer compressed sensing [10]. Finally, we provide evidence that zooming in, a strategy known to boost accuracy [40, 61], is effective because it discards uninformative information rather than displaying new details.

Our contributions are the following: **(1)** We introduce a novel black-box attribution method, dubbed wavelet scale attribution method (WCAM), based on the perturbation of the wavelet transform of an image, **(2)** we show that this attribution method not only explains where the model sees but also *what* it sees, thus paving the way for a fine-grained analysis of the failure cases of deep models under image corruptions. To demonstrate the potential of the WCAM, **(3)** we use them to explain how image corruptions fool models, and **(4)** to discuss what information a model uses for classification.

2 Literature and background

2.1 Related works

Explainability Explainability in computer vision typically quantifies the contribution of an image's pixel or region to a model's prediction. Saliency [56] was the first method to identify such regions.

The approach used the model's gradients and the classification score. A line of works improved this approach: instead of using the model's gradients, other works used the model's activations to generate explanations. It is the principle behind the class activation map (CAM, [69]), which has also been further refined [55, 52, 60]. These methods are quick to compute an explanation but require access to the model's gradients or activation. We often refer to these methods as "white-box" explanation methods. By contrast, "black-box" methods are model agnostic. Explanations are computed by perturbing (e.g., occluding parts of the image) the inputs and computing a score that reflects the model's sensitivity to the perturbation. The various proposed methods, e.g., Occlusion [68], LIME [50], RISE [48], Sobol [12], HSIC [47] or EVA [13] differ in that they use different sampling strategies to explore the space of perturbations. However, the main limitation of these methods is that they only explain *where* the model focuses and may lack faithfulness [2].

To begin addressing the *what*, [15] recently introduced CRAFT. This method combines matrix factorization for concept identification and Grad-CAM [52] for concept localization on the input image. Another line of work focused on identifying the most significant points in the training dataset through influence functions [36]. However, such approaches require access to the model and do not focus on explaining a model's decision under image corruptions. Therefore, we still need an informative explanation method to understand why image corruptions can fool deep learning models.

Robustness to natural image corruptions As image corruptions are ubiquitous in real-world applications, numerous works have tackled this issue. We can distinguish works that aim to improve model robustness and those aiming to explain the (lack of) robustness. Data augmentation emerged as a very efficient paradigm to improve model accuracy under natural image corruptions [24]. Popular methods include AutoAugment [8], AugMix, [26], PixMix [28], SIN [19]. These methods consist in implicitly regularizing the model in a given direction, e.g., to lower the texture bias and increase the shape bias in the case of SIN.

Studies explaining model sensitivity to image corruptions pointed out the crucial role of high-frequency components in the image. [63] showed that machines perceived both high-frequency and "semantic" components (*i.e.*, low-frequency components, shapes), whereas humans mostly rely on semantic components. [67] introduced Fourier Maps, which consist in perturbing the Fourier domain of images to quantify the contribution of low and high-frequency components in a model's decision. This "frequency bias" [1] found empirical and theoretical groundings in several works [49, 16]. Finally, recent results showed that robust and adversarially trained models rely less on high-frequency components than vanilla models [34, 70, 71, 6]. These studies provide a general characterization of models, *i.e.*, *model-based* explanations.

Despite being very useful for understanding the properties of robust models, robustness assessment only provides *model-based* explanations: we still need to bring these insights for an *instance-based* robustness assessment. Performing such instance-based assessment would enable us to understand why models fail to classify a given image under image corruptions correctly, thus improving our trust in these systems when deployed in practical settings [21]. Indeed, even the most robust models make prediction errors. Fourier transforms are insufficient to achieve this: they only localize the components in frequency, not space, whereas wavelets are defined spatially and in scales. We show that wavelets help close the gap in understanding *what* models see.

2.2 Background

Wavelet transform A wavelet is an integrable function $\psi \in L^2(\mathbb{R})$ with zero average, normalized and centered around 0. Unlike a sinewave, a wavelet is localized in space and the Fourier domain. It implies that dilatations of this wavelet enable to scrutinize different frequencies (scales) while translations enable to scrutinize spatial location. To compute an image's (continuous) wavelet transform (CWT), one first defines a filter bank $\mathcal{D}$ from the original wavelet ψ with the scale factor s and the 2D translation in space u. We have

$$\mathcal{D} = \left\{ \psi_{s,u}(x) = \frac{1}{\sqrt{s}} \psi \left(\frac{x-u}{s} \right) \right\}_{u \in \mathbb{R}^2,\ s \geq 0}, \tag{1}$$

where $|\mathcal{D}| = J$, and J denotes the number of levels. The computation of the wavelet transform of a function $f \in L^2(\mathbb{R})$ at location x and scale s is given by

$$\mathcal{W}(f)(x,s) = \int_{-\infty}^{+\infty} f(u) \frac{1}{\sqrt{s}} \psi^* \left(\frac{x-u}{s} \right) \mathrm{d}u, \tag{2}$$

which can be rewritten as a convolution [44]. Computing the multilevel decomposition of f requires applying Equation 2 J times with all dilated and translated wavelets of $\mathcal{D}$. [43] showed that one could implement the multilevel dyadic decomposition of the discrete wavelet transform (DWT) by applying a high-pass filter H to the original signal f and subsampling by a factor of two to obtain the *detail* coefficients and applying a low-pass filter G and subsampling by a factor of two to obtain the *approximation* coefficients. Repeating this operation on the approximation coefficients yields a multilevel transform where the j^{th} level extracts information at resolutions between 2^j and 2^{j-1} pixels. As we deal with images (2D signals), the detail coefficients can be decomposed into horizontal, vertical, and diagonal components.

Sobol sensitivity analysis Let $(X_1, \ldots, X_K)$ be independent random variables and $\mathcal{K} = \{1, \ldots, K\}$ denote the set of indices. Let f be a model, X an input, and $f(X)$ the model's decision (e.g., the output probability). We denote $f_\kappa = f_\kappa(X_\kappa)$ the partial contributions of the variables $(X_k)_{k \in \kappa}$ to the score $f(X)$. The Sobol-Hoeffding decomposition [29] decomposes the decision score $f(X)$ into summands of increasing dimension

$$f(X) = f_\emptyset + \sum_{\kappa \in \mathcal{P}(\mathcal{K}) \setminus \{\emptyset\}} f_\kappa(X_\kappa), \tag{3}$$

Where $f_\emptyset$ denotes the prediction without any features. Under the orthogonality condition $\forall (u,v) \in \mathcal{K}^2$ such that $u \neq v$, $\mathbb{E}\left[f_u(X_u) f_v(X_v)\right] = 0$, we can derive from Equation 3 the variance of the model's score

$$Var(f(X)) = \sum_{\kappa \in \mathcal{P}(\mathcal{K})} Var(f_\kappa(X_\kappa)), \tag{4}$$

Equation 4 enables us to describe the influence of a subset κ of features as the ratio between its own and total variance. This corresponds to the first order **Sobol index** given by

$$S_\kappa = \frac{Var(f_\kappa(X_\kappa))}{Var(f(X))}. \tag{5}$$

S_κ measures the proportion of the output variance $Var(f(X))$ explained by the subset of variables X_κ [59]. In particular, S_k only captures the *direct* contribution of the feature X_k to the model's decision. To capture the indirect effect, due to the effect of X_k on the other variables, **Sobol total indices** S_{T_k} [30] can be computed as

$$S_{T_k} = \sum_{\kappa \in \mathcal{P}(\mathcal{K}),\, k \in \kappa} S_\kappa. \tag{6}$$

Sobol total indices (STIs) measure the contribution of the k^{th} feature, taking into account both its *direct* effect and its *indirect* effect through its interactions with the other features.

Efficient estimation of Sobol indices As seen from the definition of the Sobol index, estimating the impact of a feature k on the model's decision requires recording the partial contribution $f_k(X_k)$. This partial contribution corresponds to a *forward*. Estimating Sobol indices requires computing variances by drawing at least N samples and computing N forwards to estimate a first-order Sobol index S_k of a single feature k. As we are interested in the STI of a feature k, we need to estimate the Sobol index of all sets of features $\kappa \in \mathcal{K}$ such that $k \in \kappa$. To minimize the computational cost of this computation, [12] introduced an efficient sampling strategy based on Quasi-Monte Carlo methods [46] to generate the N perturbations of dimension K applied to the input and used Jansen's estimator [32] to estimate the STIs given the models' outputs and the quasi-random perturbations. Their approach requires $N(K+2)$ forwards [12].

To estimate the STIs, they draw two matrices from a Quasi-Monte Carlo sequence of size $N \times K$ and convert them into perturbations, which we apply to X. The perturbated input yields two matrices A and B. a_{jk} (resp. b_{kj}) is the element of A (resp. B) corresponding to the k^{th} feature and the

j^{th} sample. For the k^{th} feature, they define $C^{(k)}$ in the same way as A, except that the column corresponding to feature k is replaced by the column of B. They then derive an empirical estimator for the Sobol index and STI as

$$\hat{S}_k = \frac{\hat{V} - \frac{1}{2N}\sum_{j=1}^{N}\left[f(B_j) - f\left(C_j^{(k)}\right)\right]^2}{\hat{V}} \qquad \hat{S}_{T_k} = \frac{\frac{1}{2N}\sum_{j=1}^{N}\left[f(A_j) - f\left(C_j^{(k)}\right)\right]^2}{\hat{V}} \tag{7}$$

where $f_\emptyset = \frac{1}{N}\sum_{j=1}^{N} f(A_j)$ and $\hat{V} = \frac{1}{N-1}\sum_{j=0}^{N}\left[f\left(A_j\right) - f_\emptyset\right]^2$,. Further implementational details can be found in [12].

3 Methods

3.1 Wavelet scale attribution method (WCAM)

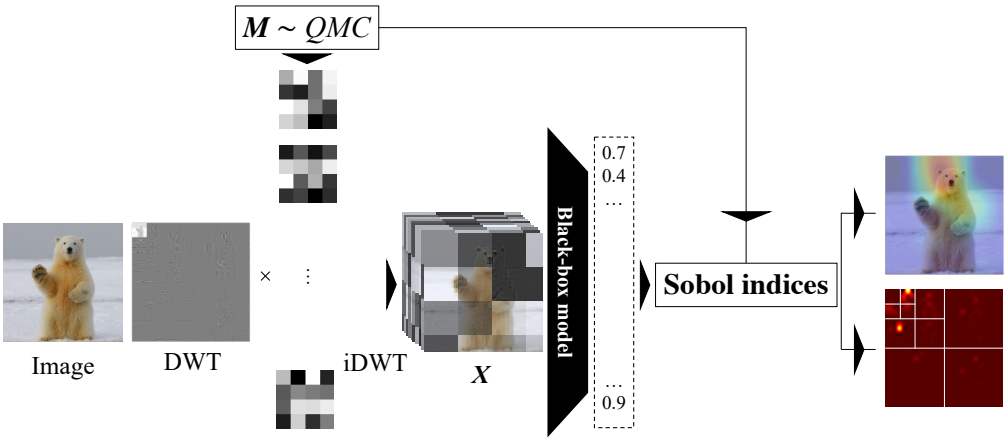


Figure 2: Flowchart of the wavelet scale attribution method (WCAM).

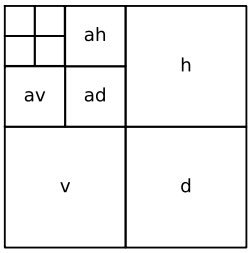
The **W**avelet s**C**ale **A**ttribution **M**ethod (**WCAM**) is an attribution method that quantifies the importance of the components of the *wavelet transform* of an image to the predictions of a model. Figure 2 depicts the principle of the WCAM. The importance of the regions of the wavelet transform of the input image is estimated by **(1)** generating masks from a Quasi-Monte Carlo sequence, **(2)** evaluating the model on perturbed images. We obtain these images by computing the DWT of the original image, applying the masks on the DWT to obtain perturbed DWT, and inverting the perturbed DWT to generate perturbed images. For a single image, we generate $N(K+2)$ perturbed images[1] **(3)** We estimate the total Sobol indices of the perturbed regions of the wavelet transform using the masks and the model's outputs using Jansen's estimator [32]. [12] introduced this approach to estimate the importance of image regions in the pixel space. We generalize it to the wavelet domain.

Scales Generally, scales are indexed in pixels: the last (or outermost) level corresponds to scales ranging from 1 to 2 pixels, and the second from 2 to 4 pixels. The approximation coefficients (upper-left corner of the image) correspond to scales *above* $2^{\texttt{levels}}$ pixels. The localization in scale of important components enables a detailed interpretation of the important attributes that contribute to the prediction, as demonstrated in figure 3b

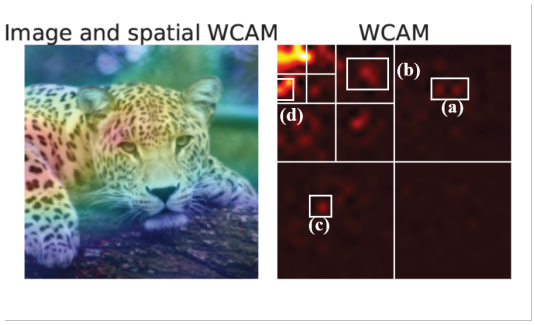
The WCAM expands attribution to the space-scale domain The main contribution of the WCAM is to decompose attribution in terms of scales, allowing for a straightforward interpretation as structural elements on the image (fine-grained or coarse textures, shapes). Indeed, the wavelet transform decomposes an image into components of different scales and, for each scale, into vertical, diagonal, and horizontal components (see Figure 3a). As depicted in Figure 3b, the WCAM unwraps

[1] On an RGB image, we apply the DWT channel-wise and apply the same perturbation to each channel.

the important scales for a given location. For instance, we can see that the eyes are important at different scales: 1-2-pixel scale (region **(a)**), corresponding to the finest details of the eyes, and 2-4-pixel scale (region **(b)**), corresponding to the contour of the eyes. The fur is also important at the 1-2-pixel scale (region **(c)**), which corresponds to the texture of the fur, and at the 4-8-pixel scale (region **(d)**) corresponding to the points on the paw.



(a) Labels of the regions of the wavelet transform.



(b) Spatial-CAM (left) and WCAM (right). The brighter, the most important for prediction.

Figure 3: Example of a WCAM (Figure 3b) and the labels within the wavelet transform (Figure 3a).

Spatial WCAM We derive the spatial WCAM by summing the Sobol indices through the levels. We provide more details in appendix A.2.1. The spatial WCAM boils down to a usual attribution method in the pixel domain. As further discussed in appendix A.2.2, we evaluated the spatial WCAM on the Xplique benchmark [14] and showed that it is competitive with existing approaches.

Fourier attribution maps To bridge the gap with previous methods that leveraged perturbation in the Fourier domain to evaluate model robustness, we also introduced Fourier activation maps (FAM). FAMs share the same principle as the WCAM, except that we apply the perturbation to the amplitude of the Fourier transform of the image. FAMs highlight the frequencies that contribute the most to a model's prediction, as done in previous works such as [71] and [6]. Besides, the WCAM and FAM behave similarly regarding the quantification of the importance of the frequency components. We focus on the WCAM as it enables a more throughout analysis of the behavior of the models. We refer the interested reader to appendix A.3 for more details on implementing the FAM and our replication of prior studies using our approach.

3.2 Unveiling why image corruptions affect model's predictions using the WCAM

Models The model testbench comprises convolutional neural networks (CNNs), embodied by the ResNet-50 [23] model and the vision transformers, and more specifically, the ViT-B16 architecture, introduced by [11]. We consider six popular training strategies. Following [7], we distinguish three training families: the "standard training" (ST) corresponding to the off-the-shelf use of a ResNet-50 loaded from Torchvision, the "Robust training" (RT) approaches (AugMix [26], PixMix [28] and SIN [20]), and the "adversarial training" (AT) (standard adversarial, [42], the fast training approach [64], and the "free" method [53]). Robust and adversarial methods are implemented on a ResNet-50 backbone.

Datasets We focus on ImageNet [51] and ImageNet-C [25]. ImageNet-C features common image perturbations such as blur, pixelation, `.jpeg` compression, and defocus blur that can occur in real life. Strengths levels, ranging from 1 to 6, can be applied to the perturbations. The higher the strength, the stronger the perturbation. When needed, we generate corrupted image samples using ImageNet-C source code. In appendix B.2, we apply the WCAM to other datasets and demonstrate its ability to troubleshoot a broader scope of distribution shifts (*, e.g.*, renditions [24] and natural adversarial examples [27]). We do not consider datasets such as CIFAR-10-100 [37] or TinyImageNet [38] as the image size of these datasets is too small to decompose the image above one level meaningfully.

Experimental details We discard the approximation coefficients. These coefficients, corresponding to the context image (upper left corner of the wavelet transform), can no longer be interpreted in terms of scale as they will contain details with a scale greater than $2^{\texttt{levels}}$ pixels. We use three decomposition levels, which are sufficient to decompose the model's predictions through different scales. We carried out our main experiments using up to 3 Nvidia Titan Xp GPUs and the Xplique [14] benchmarks on a Google Colaboratory [4] GPU cluster.

4 Results

4.1 Understanding how image corruptions affect classification models

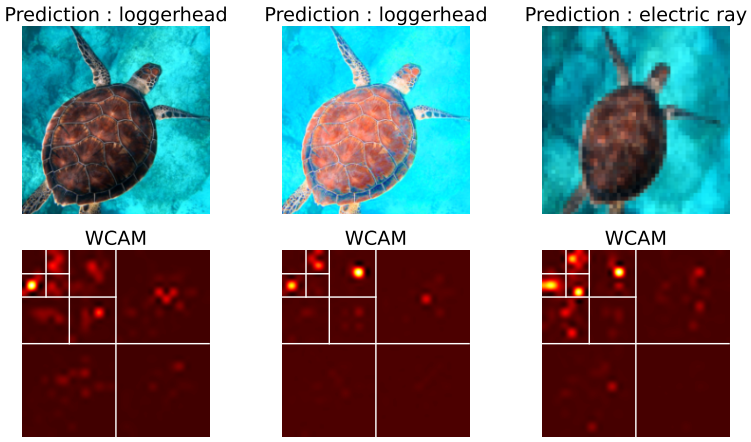


Figure 4: Localization in space and scale of the important coefficients for the prediction, as highlighted by the WCAM when a vanilla ResNet-50 [23] faces an image corruption coming from ImageNet-C [25]. The leftmost column plots the original image, the center image a corrupted image correctly predicted, and the rightmost image a corrupted image incorrectly predicted.

Models rely on a few coefficients well defined in space and scale Figure 4 plots the WCAMs of the baseline model's prediction for clean and corrupted images. Overall, we can see that the model (a vanilla ResNet-50) relies on a few coefficients well-defined in space and scale to make its prediction. Higher scale coefficients (*i.e.*, high frequencies) contribute the least. Even at larger scales, the model focuses on a few hot spots. We provide additional examples in appendix B.5.1.

Natural image corruptions mainly alter important scales, not localizations When comparing the WCAMs obtained across corrupted images, we can see that corruptions alter the important scales but that the *spatial* localizations remain centered around the object. Notably, details at the 1-pixel scale are more important for the prediction on the rightmost image: the pixelation may have led the model to mistake the back of the turtle for the back of an electric ray. The fact that perturbations appear near the object could explain why attribution in the pixel domain is not very informative for debugging.

A consistent behavior across baselines We can summarize the behavior illustrated by Figure 5 as follows: **(1)** models rely on a few wavelet components in the space-scale domain and **(2)** image corruptions target scales, and are thus invisible for standard attribution methods. We computed the WCAM across various models and images to see how universal this behavior was. Figure 5 plots an example for two methods, SIN [20] and the standard adversarial method [42]. We can see that these models behave according to the same qualitative pattern as the vanilla ResNet-50. We provide additional examples in appendix B.5.2.

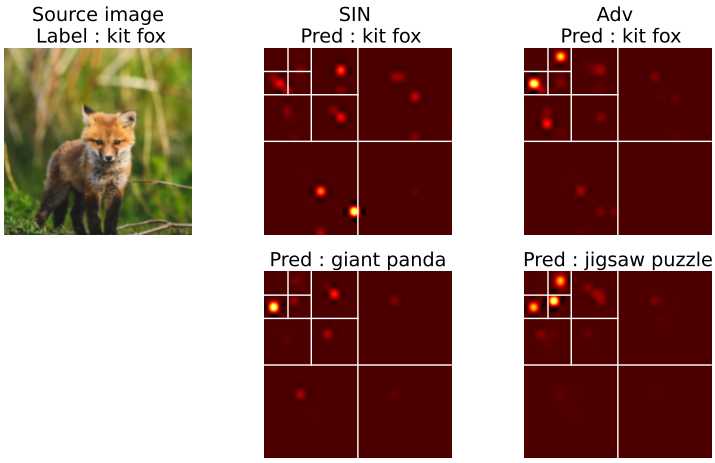


Figure 5: Original image (left) and corresponding WCAMs for a clean (upper row) and corrupted (lower row) version across two baselines (SIN [20], middle column and Adv [42], left column).

4.2 Identifying the sufficient information for prediction

In this section, we leverage results from section 4.1 to generate sufficient images, *i.e.*, images that contain only the information necessary for correct classification.

Sufficient image We call *sufficient image* the image reconstructed from the n first wavelet coefficients according to their corresponding Sobol indices such that the model can correctly predict the image's label. Figure 6 displays examples of such images. In our examples, we can see that for the image of a cat, the model needs detailed information around the eyes, which is not the case for the fox. In both cases, we also see that the models do not need information from the background, as we can completely hide it without changing the prediction. Identifying a sufficient image could have numerous applications, for instance, in transfer compressed sensing [10]. In some applications (e.g., medical imaging), data acquisition is expensive [10]. Using the information provided by the WCAM, we can learn how to compress signals to preserve the essential information for classification.

Reconstruction depth As the number of coefficients necessary to define the sufficient image varies from one image to the other, we wondered whether the *reconstruction depth* (*i.e.*, the number of coefficients needed to reconstruct a sufficient image) was predictive of the robustness of the prediction. The mechanism is the following: the more coefficients needed to reconstruct the sufficient image, the higher the dispersion in the scale-space domain. The higher the dispersion, the higher the chances of disruption due to a corruption. Implicitly, we assume that the model recovers information from the largest scales (*i.e.*, low frequencies, more robust) to the smallest (*i.e.*, high frequencies, less robust). The literature on the spectral bias [49, 33, 66, 65] showed that models learn information in such a structured way.

For this task, we define the robustness of a prediction as the share of images correctly predicted over the complete set of ImageNet-C corruptions of an image. We compute the reconstruction

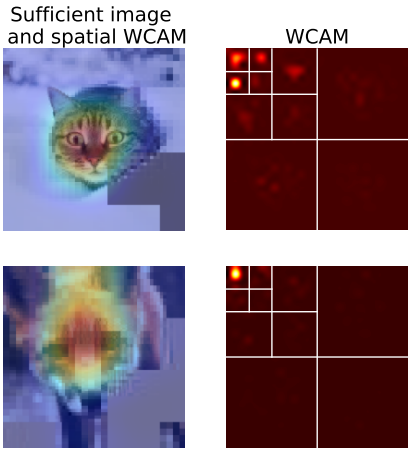


Figure 6: Sufficient images reconstructed from the WCAM.

depth and the robustness of the prediction for a sample of ImageNet images. In appendix B.3, we highlight a pattern between the robustness of a prediction and the reconstruction depth. Reconstruction depth could pave the way toward measures of the robustness of a prediction, which can be very useful for flagging unreliable model predictions.

4.3 Scale consistency: Do models use new information from zoomed-in images?

Zooming-in is efficient to improve model accuracy [40]. It discards elements from the background and increases the number of pixels that describe the object of interest [40, 61]. However, whether gains rest on new information or less background noise remains unclear. We quantified the amount of "new" information brought by the zoom using the WCAM as follows: we evaluate the WCAM on original images and zoomed-in images and compute the importance (in terms of Sobol indices) of each level. We use a four-level decomposition for the zoomed-in images: the last level corresponds to invisible details on the regular image. The penultimate level on the zoomed-in corresponds to the last level on the original image. The importance of the last level of the zoomed-in image will indicate the reliance of models on this previously invisible information. We provide further implementational details in appendix B.4.1.

Classification models are zoom-consistent As seen from Table 1, the importance of the level introduced by the zoom is negligible. As such, the higher accuracy comes from the fact that there is less background on the image, not because the model finds new features. We can also see that adversarial models rely less on high frequencies (or higher levels) than robust models. We provide the full table in appendix B.4.2.

Table 1: **Importance** (quantified by the Sobol indices of the WCAM) of each scale level in the prediction for regular **(Reg.)** and zoomed-in ($\times 2$) transforms of 100 images sampled from ImageNet. When we zoom in on the image, we decompose it into four levels. The highest level (4) contains information at scales that can only be seen on the zoomed-in image. The bolded value indicates the share of the *new* information leveraged by the model to classify the image. Columns "Robust" and "Adversarial" report the means across the models from these classes. Standard errors in parenthesis.

Level	*Baseline* [23]		*Robust* [26, 28, 20]		*Adversarial* [42, 53, 64]		*ViT* [11]	
	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$
0	0.837	0.752	0.895	0.869	0.954	0.906	0.830	0.611
	(0.064)	(0.137)	(0.082)	(0.101)	(0.033)	(0.080)	(0.075)	(0.164)
1	0.130	0.190	0.077	0.095	0.039	0.080	0.137	0.295
	(0.053)	(0.107)	(0.063)	(0.077)	(0.029)	(0.071)	(0.063)	(0.132)
2	0.028	0.047	0.023	0.030	0.006	0.013	0.029	0.078
	(0.012)	(0.030)	(0.024)	(0.033)	(0.005)	(0.011)	(0.018)	(0.043)
3	0.005	0.010	0.004	0.005	0.001	0.001	0.004	0.014
	(0.002)	(0.006)	(0.005)	(0.005)	(0.001)	(0.001)	(0.002)	(0.008)
4	(-)	**0.001**	(-)	**0.001**	(-)	**0.000**	(-)	**0.002**
	(-)	(0.001)	(-)	(0.001)	(-)	(0.000)	(-)	(0.001)

5 Conclusions and future work

We introduce the **W**avelet s**C**ale **A**ttribution **M**ethod (**WCAM**), a generalization attribution to the space-scale domain. The WCAM highlights the important regions in the scale-space domain using perturbations efficient perturbations of the wavelet transform of the input image. We estimate the contribution of the regions of the wavelet transform using total Sobol indices. The WCAM explains *where* and *what* models see as they decompose the important regions in space and scale. They are informative for explaining the sensitivity of vision models to image corruptions. We demonstrate the potential of the WCAM for model debugging and failure cases analysis as we showed how image corruptions mislead vision models (robust, non-robust, adversarial, and vision transformers): corruptions disrupt the important *scales*, and not the important locations for prediction. We also showed that the WCAM identified the sufficient amount of information a model needs to make a

prediction. Finally, we showed that zoom-in increases classification accuracy because it discards background noise, not because it brings new details of the object.

Limitations The main limitation of our approach is that it is computationally more expensive than existing black-box attribution methods. The second limit is that it mainly focuses on a qualitative assessment of the model's prediction. We explored quantification with the reconstruction depth but still have to find a convincing metric.

Future works We plan to discuss the applicability of WCAM in a real use case in remote sensing [35], with explicit scales. We also wish to overcome the quantification problem by linking the WCAM and out-of-distribution detection. Finally, we intend to discuss whether the WCAM helps explain potentially harmful biases, e.g., due to the biases in ImageNet [54] for ImageNet-trained classifiers.

6 Acknowledgements

This work is funded by RTE France, the French transmission system operator, and benefited from CIFRE funding from the ARNT. The authors gratefully acknowledge the support of this project. The authors would like to thank Thomas Fel, the first author of the original article *"Look at the Variance! Efficient Black-box Explanations with Sobol-based Sensitivity Analysis"* for his insightful comments and advice while preparing this work. The authors would also like to thank Hugo Thimonier for his helpful advice and feedback and Thomas Heggarty for proofreading the final manuscript.

References

[1] Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the High-Frequency Bias in Convolutional Neural Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 863–871, Nashville, TN, USA, June 2021. IEEE.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 9505–9515. Curran Associates, Inc., 2018.

[3] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and Aggregating Feature-based Model Explanations, May 2020. arXiv:2005.00631 [cs, stat].

[4] Ekaba Bisong. Google Colaboratory. In Ekaba Bisong, editor, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pages 59–64. Apress, Berkeley, CA, 2019.

[5] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*, 8(8):832, July 2019.

[6] Yiting Chen, Qibing Ren, and Junchi Yan. Rethinking and Improving Robustness of Convolutional Neural Networks: a Shapley Value-based Approach in Frequency Domain. October 2022.

[7] Zhiming Chen, Wei Xue, Weiwei Tian, Yunhua Wu, and Bing Hua. Toward deep neural networks robust to adversarial examples, using augmented data importance perception. *Journal of Electronic Imaging*, 31(6):063046, December 2022. Publisher: SPIE.

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data, April 2019. arXiv:1805.09501 [cs, stat].

[9] Hervé Delseny, Christophe Gabreau, Adrien Gauffriau, Bernard Beaudouin, Ludovic Ponsolle, Lucian Alecu, Hugues Bonnin, Brice Beltran, Didier Duchel, Jean-Brice Ginestet, Alexandre Hervieu, Ghilaine Martinez, Sylvain Pasquet, Kevin Delmas, Claire Pagetti, Jean-Marc Gabriel, Camille Chapdelaine, Sylvaine Picard, Mathieu Damour, Cyril Cappi, Laurent Gardès, Florence De Grancey, Eric Jenn, Baptiste Lefevre, Gregory Flandin, Sébastien Gerchinovitz, Franck Mamalet, and Alexandre Albore. White Paper Machine Learning in Certified Systems, March 2021. arXiv:2103.10529 [cs]version: 1.

[10] Manik Dhar, Aditya Grover, and Stefano Ermon. Modeling Sparse Deviations for Compressed Sensing using Generative Models, July 2018. arXiv:1807.01442 [cs, stat].

[11] 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, June 2021. arXiv:2010.11929 [cs].

[12] Thomas Fel, Remi Cadene, Mathieu Chalvidal, Matthieu Cord, David Vigouroux, and Thomas Serre. Look at the Variance! Efficient Black-box Explanations with Sobol-based Sensitivity Analysis, November 2021. arXiv:2111.04138 [cs].

[13] Thomas Fel, Melanie Ducoffe, David Vigouroux, Remi Cadene, Mikael Capelle, Claire Nicodeme, and Thomas Serre. Don't Lie to Me! Robust and Efficient Explainability with Verified Perturbation Analysis, March 2023. arXiv:2202.07728 [cs].

[14] Thomas Fel, Lucas Hervier, David Vigouroux, Antonin Poche, Justin Plakoo, Remi Cadene, Mathieu Chalvidal, Julien Colin, Thibaut Boissin, Louis Bethune, Agustin Picard, Claire Nicodeme, Laurent Gardes, Gregory Flandin, and Thomas Serre. Xplique: A Deep Learning Explainability Toolbox, June 2022. arXiv:2206.04394 [cs].

[15] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept Recursive Activation FacTorization for Explainability, March 2023. arXiv:2211.10154 [cs].

[16] Sara Fridovich-Keil, Raphael Gontijo-Lopes, and Rebecca Roelofs. Spectral Bias in Practice: The Role of Function Frequency in Generalization. October 2022.

[17] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. arXiv:2004.07780 [cs, q-bio].

[18] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *arXiv preprint arXiv:2106.07411*, 2021.

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[20] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. April 2023.

[21] Misgina Tsighe Hagos, Kathleen M. Curran, and Brian Mac Namee. Identifying Spurious Correlations and Correcting them with an Explanation-based Learning, December 2022. arXiv:2211.08285 [cs].

[22] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, December 1960.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. pages 8340–8349, 2021.

[25] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, March 2019. arXiv:1903.12261 [cs, stat].

[26] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty, February 2020. arXiv:1912.02781 [cs, stat].

[27] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural Adversarial Examples, March 2021. arXiv:1907.07174 [cs, stat].

[28] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures, March 2022. arXiv:2112.05135 [cs].

[29] Wassily Hoeffding. A Class of Statistics with Asymptotically Normal Distribution. In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics: Foundations and Basic Theory*, Springer Series in Statistics, pages 308–334. Springer, New York, NY, 1992.

[30] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17, April 1996.

[31] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, April 2017. arXiv:1704.04861 [cs].

[32] Michiel J. W. Jansen. Analysis of variance designs for model output. *Computer Physics Communications*, 117(1):35–43, March 1999.

[33] Jason Jo and Yoshua Bengio. Measuring the tendency of CNNs to Learn Surface Statistical Regularities, November 2017. arXiv:1711.11561 [cs, stat].

[34] Nikos Karantzas, Emma Besier, Josue Ortega Caro, Xaq Pitkow, Andreas S. Tolias, Ankit B. Patel, and Fabio Anselmi. Understanding robustness and generalization of artificial neural networks through Fourier masks, March 2022. arXiv:2203.08822 [cs, eess].

[35] Gabriel Kasmi, Yves-Marie Saint-Drenan, David Trebosc, Raphaël Jolivet, Jonathan Leloux, Babacar Sarr, and Laurent Dubus. A crowdsourced dataset of aerial images with annotated solar photovoltaic arrays and installation metadata. *Scientific Data*, 10(1):59, January 2023.

[36] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions, December 2020. arXiv:1703.04730 [cs, stat].

[37] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

[38] Ya Le and Xuan Yang. Tiny ImageNet Visual Recognition Challenge. 2015.

[39] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3), September 2015. arXiv:1511.01644 [cs, stat].

[40] Xiao Li, Jianmin Li, Ting Dai, Jie Shi, Jun Zhu, and Xiaolin Hu. Rethinking Natural Adversarial Examples for Classification Models, February 2021. arXiv:2102.11731 [cs].

[41] Xiaodan Li, Yuefeng Chen, Yao Zhu, Shuhui Wang, Rong Zhang, and Hui Xue. ImageNet-E: Benchmarking Neural Network Robustness via Attribute Editing, March 2023. arXiv:2303.17096 [cs].

[42] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, September 2019. arXiv:1706.06083 [cs, stat].

[43] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[44] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[45] M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245, 1979. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

[46] William J. Morokoff and Russel E. Caflisch. Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122(2):218–230, December 1995.

[47] Paul Novello, Thomas Fel, and David Vigouroux. Making Sense of Dependence: Efficient Black-box Explanations Using Dependence Measure, September 2022. arXiv:2206.06219 [cs, stat].

[48] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models, September 2018. arXiv:1806.07421 [cs].

[49] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks, May 2019. arXiv:1806.08734 [cs, stat].

[50] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016. arXiv:1602.04938 [cs, stat].

[51] 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, January 2015. arXiv:1409.0575 [cs].

[52] 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, February 2020. arXiv:1610.02391 [cs].

[53] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial Training for Free!, November 2019. arXiv:1904.12843 [cs, stat].

[54] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D. Sculley. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World, November 2017. arXiv:1711.08536 [stat].

[55] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences, April 2017. arXiv:1605.01713 [cs].

[56] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, April 2014. arXiv:1312.6034 [cs].

[57] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015. arXiv:1409.1556 [cs].

[58] I. M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, January 1967.

[59] Ilya Meerovich Sobol. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe modelirovanie*, 2(1):112–118, 1990. Publisher: Russian Academy of Sciences, Branch of Mathematical Sciences.

[60] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks, June 2017. arXiv:1703.01365 [cs].

[61] Mohammad Reza Taesiri, Giang Nguyen, Sarra Habchi, Cor-Paul Bezemer, and Anh Nguyen. Zoom is what you need: An empirical study of the power of zoom and spatial biases in image classification, April 2023. arXiv:2304.05538 [cs].

[62] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, September 2020. arXiv:1905.11946 [cs, stat].

[63] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8681–8691, Seattle, WA, USA, June 2020. IEEE.

[64] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training, January 2020. arXiv:2001.03994 [cs, stat].

[65] Zhi-Qin John Xu, Yaoyu Zhang, and Tao Luo. Overview frequency principle/spectral bias in deep learning, October 2022. arXiv:2201.07395 [cs].

[66] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks. *Communications in Computational Physics*, 28(5):1746–1767, June 2020. arXiv:1901.06523 [cs, stat].

[67] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision, September 2020. arXiv:1906.08988 [cs, stat].

[68] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks, November 2013. arXiv:1311.2901 [cs].

[69] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[70] Jiajin Zhang, Hanqing Chao, Amit Dhurandhar, Pin-Yu Chen, Ali Tajer, Yangyang Xu, and Pingkun Yan. When Neural Networks Fail to Generalize? A Model Sensitivity Perspective, December 2022. arXiv:2212.00850 [cs].

[71] Zhuang Zhang, Dejian Meng, Lijun Zhang, Wei Xiao, and Wei Tian. The range of harmful frequency for DNN corruption robustness. *Neurocomputing*, 481:294–309, April 2022.

A Implementational details on the WCAM and FAM

A.1 Wavelet scale attribution method

A.1.1 Generation of the masks

To generate the masks, we follow the sampling procedure introduced by [12]. Their approach consist in drawing two independent matrices of size $N \times K$ from a Sobol low discrepancy sequence. N is the number of designs, necessary to estimate the variance and K is the dimension of the sequence.

We reshape this sequence as a two-dimensional mask to generate our perturbation. By default, we perturb the wavelet transform with a mask of size 28×28, so as to balance between the dimensionality of the sequence and the accuracy of the perturbation. We reshape the 784-dimensional sequence to a grid of 28×28 to define our perturbation masks. We tried alternative mapping from the unidimensional sequence to the mask, but with limited effect on the dimensionality reduction and at the expense of the meaningfulness of the perturbation in the wavelet domain. Figure 7 depicts an illustration of our workflow for one mask to generate the images that are then passed to the model.

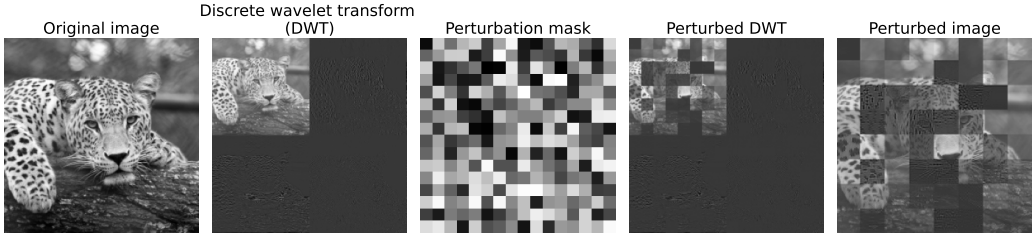


Figure 7: Workflow on a grayscale image and for a 1-level wavelet transform. We first compute the discrete wavelet transform of the image, then apply a mask on the DWT. This yields the perturbed DWT which we invert to generate the perturbed image. This image is then passed to the model.

A.1.2 Effect of the grid size, number of designs and sampler on the explanations

To evaluate the effect of the parameters `grid_size` and `nb_design` and of the samplers on the estimation of the Sobol indices, we consider 100 images sampled from ImageNet and compute the Insertion [48] and Deletion [48] scores. These pointing metrics measure the accuracy of an explanation. We compute these scores on the *spatial* WCAM to study the effect of the parameters. Benchmarks were carried out using the Xplique toolbox [14].

Evaluation metrics We evaluate the accuracy of our explanations based on two metrics, introduced by [48]. The first one is deletion. Deletion measures the drop in probability in the predicted probability of a class when removing the pixels highlighted by the explanation. The higher the drop, the better the explanation. At step t, the u most important variables according to deletion are given by:

$$\text{Del}^{(t)} = f\left(x_{x_u = x_0}\right) \tag{8}$$

Where x_0 is the baseline set, which is set to 0 in the Xplique library.

On the other hand, the insertion measures the contribution of the pixels highlighted by the explanation to the predicted probability when *inserting* a feature. The higher the increase, the better the explanation. At step t, the insertion score for the u most important features is given by:

$$\text{Ins}^{(t)} = f\left(x_{x_{\bar{u}} = x_0}\right) \tag{9}$$

Where x_0 is the same baseline state as deletion. Insertion and deletion are area-under-curve metrics, the lower the deletion the better, and the higher the insertion the better. The intuition is as follows: if an explanation picks the most important features (according to the probability score), then we should

remove (resp. insert) the most important feature at step $t = 1$, the second most important at step $t = 2$, etc. Therefore, the probability drop (resp. increase) is the highest at the first step and then gradually decreases.

Parameters The `grid_size` parameter defines how coarse the explanation will be. On the wavelet transform, each pixel corresponds to a wavelet coefficient. A grid size that matches the input size will therefore explain all wavelet coefficients individually. However, the computational cost of such an explanation is prohibitive. Therefore, we explain *sets* of wavelet coefficients as the grid size is lower than the input size. To correctly explore the variability in space, our grid size should not be too large. We chose a grid size of 28×28 on a 224×224 input image after empirical investigation. Additionally, we test values ranging from 8 to 32, 8 corresponding to the default value of the Sobol Attribution Method [12] and 32 allowing for a 4 level decomposition.

The parameter `nb_design` corresponds to the number of deviations from the means that are used to estimate the variance of the Sobol indices. In theory, this parameter should be as high as possible for an accurate estimation of the variance. In practice, this increases the number of forwards and hence the computation time of the Sobol total indices. As the implementation of the spectral attributions require more operations than the Sobol attributions, we want to keep the number of computations and of forwards as low as possible. A straightforward way to doing it is to lower the number design.

Grid size and number of designs Figure 8 depicts the insertion and deletion scores as a function of the `grid_size` and `nb_design`. We can notice that the highest values for insertion are obtained for a grid size of 16. This is explained by the fact that we evaluate these scores on the spatial WCAM. This evaluates the quality of the "standard" explanation, without taking into account the spectral dimension of the explanations.

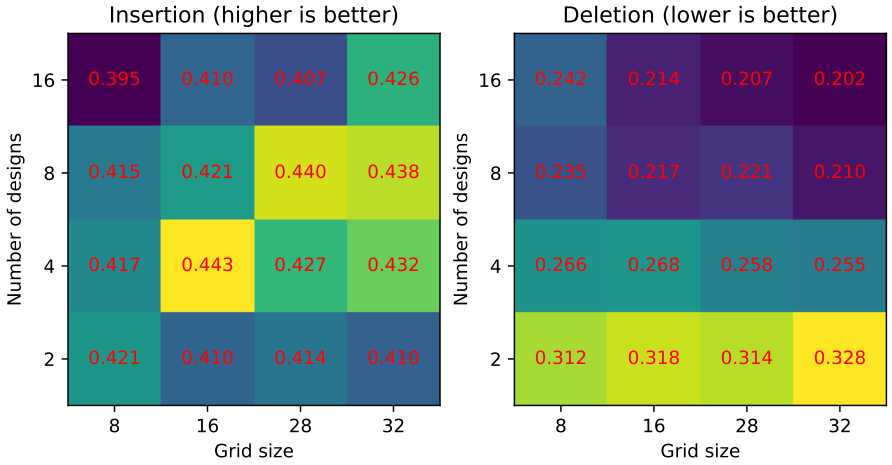


Figure 8: Insertion and deletion scores on the WCAM when the `grid_size` and `nb_design` parameters vary. For insertion, the higher the better. For deletion, the lower the better. Experiment carried out with a ResNet-50 [23] backbone on 100 randomly sampled images from ImageNet.

Visual inspection To further calibrate the `nb_design`, we estimated the WCAM using an increasing number of designs (columns) for different images (rows). We also report the computation time in seconds. Figure 9 plots the results. We can see that setting the number of designs to 8 is empirically sufficient for a consistent estimation of the Sobol indices. This number balances between accuracy and computation time.

Samplers Table 2 reports the insertion and deletion scores when the sampler changes. The baseline ScipySobolSequence gives the best results for insertion and the Halton sequence performs slightly better for deletion.

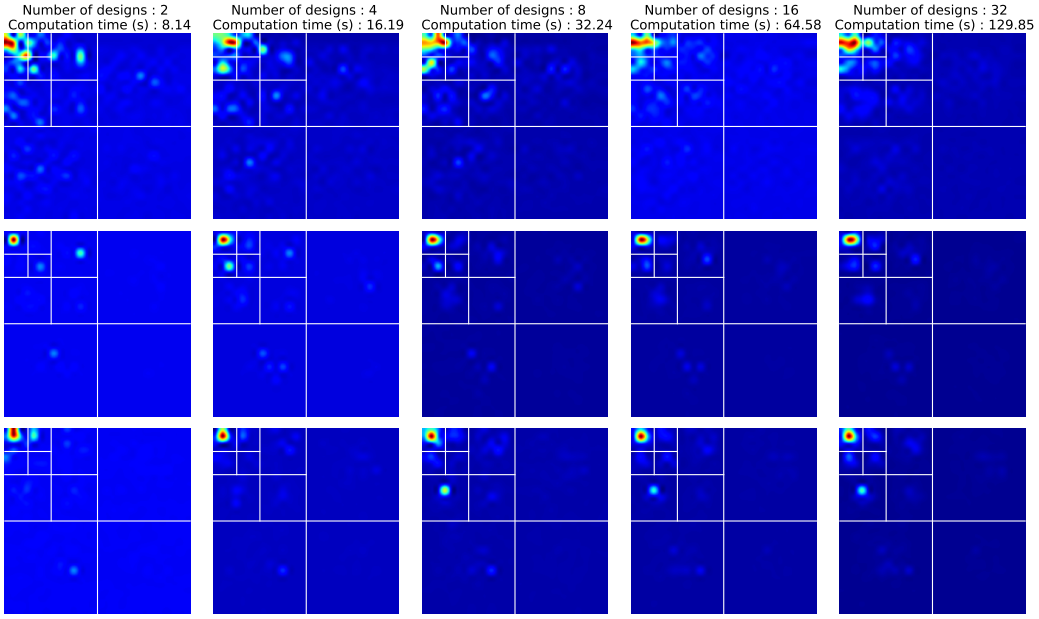


Figure 9: Effect of the number of designs on the estimation of the Sobol total indices. The higher the number of designs, the better the estimation of the conditional variance of the Sobol indices but also the higher the computational cost as the number of required inferences increases. To balance between accuracy and computational time, we can see that setting the number of designs to 8 is a reasonable choice. Each row depicts a different image. Each column plots the WCAM with a given number of designs (2, 4, 8, 16 and 32).

Sampler	Insertion (↑)	Deletion (↓)
ScipySobolSequence [58]	**0.440**	0.221
Halton [22]	0.438	**0.211**
Latin Hypercube [45]	0.423	0.217
MonteCarlo	0.435	0.223

Table 2: Effect of the sampler on the estimation of the Sobol coefficients. The `grid_size` is set to 28 and the number of designs to 8. The model backbone is a ResNet-50 [23].

A.2 WCAM as an attribution method

A.2.1 Spatial WCAM

To recover the spatial WCAM, we simply sum the Sobol indices at different scales. Each subset of the wavelet transform ("h","v", "d", "ah", "av", "ad") on Figure 10 is indexed in space. Therefore, a point in the center of the "ad" square has the same spatial localization than a point in the center of the "h" or "v" square.

The accuracy of the spatial cam rests on the definition of the Sobol coefficients devoted in estimating the importance of the approximation coefficients. With a 28×28 grid, there are 49 coefficients (grid size of 7) that describe this level, which is a little bit less that the default resolution of the baseline Sobol attribution method (which has a grid size of 8).

A.2.2 Benchmarking results

μ-Fidelity In addition to insertion and deletion, defined in section A.1.2, we evaluate the methods using a third metric, the μ-Fidelity, introduced by [3]. Contrary to insertion and deletion, which are area-under-curve metrics, the μ-Fidelity is a correlation metric. It measures the correlation between between the decrease of the predicted probabilities when features are in a baseline state and the

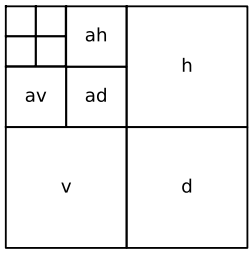


Figure 10: Decomposition of the regions of a three-level dyadic wavelet transform

importance of these features. We have:

$$\mu\text{-Fidelity} = \underset{\substack{u \subseteq \{1,\dots,K\}, \\ |u|=d}}{\text{Corr}} \left(\sum_{i \in u} g(x_i), f(x) - f(x_{x_u = x_0}) \right) \tag{10}$$

where g is the explanation function (*i.e.*, the explanation method) which quantifies the importance of the set of features u.

Results Table 3 reports the results of the comparisons of the explanations reported by the spatial WCAM compared to existing approaches. Benchmarks were carried out using the Xplique toolbox [14] on validation images from ImageNet [51].

We can see that our approach performs favorably in the case of the insertion metric: our method beats white-box attribution methods. For deletion and μ-fidelity, the performance is comparable to existing approaches. We raise awareness on the fact that we completely discard the spectral part of our explanation and that our parameterization of our method is suboptimal (see Figure 8 for a benchmarking of the hyperparameters). Therefore, this performance should be interpreted as a worst-case performance or a sanity check.

Table 3: **Deletion**, **Insertion** and μ**-Fidelity** scores obtained on 100 ImageNet validation set images. For Deletion, lower is better and for Insertion and μ-Fidelity, higher is better. Best results are **bolded** and second best underlined. All benchmarks are carried out using the Xplique library [14].

	Method	*VGG16* [57]	*ResNet50* [23]	*MobileNet* [31]	*EfficientNet* [62]
Deletion (↓)					
White-box	Saliency [56]	0.100	0.124	0.096	0.096
	Grad.-Input [55]	0.050	0.083	0.053	0.076
	Integ.-Grad. [60]	**0.041**	**0.071**	**0.045**	**0.069**
	GradCAM++ [52]	0.110	0.183	0.091	0.154
	VarGrad [52]	0.148	0.176	0.087	0.147
Black-box	RISE [48]	**0.105**	**0.143**	**0.093**	0.114
	Sobol [12]	0.110	0.144	0.097	**0.101**
	WCAM (ours)	0.178	0.221	0.173	0.185
Insertion (↑)					
White-box	Saliency [56]	0.219	0.232	0.188	0.164
	Grad.-Input [55]	0.140	0.134	0.119	0.082
	Integ.-Grad. [60]	0.171	0.170	0.186	0.147
	GradCAM++ [52]	**0.399**	**0.448**	0.084	**0.257**
	VarGrad [52]	0.223	0.257	**0.371**	0.178
Black-box	RISE [48]	**0.460**	**0.517**	**0.457**	**0.402**
	Sobol [12]	0.377	0.428	0.351	0.294
	WCAM (ours)	0.331	0.440	0.326	0.316
μ-Fidelity (↑)					
White-box	Saliency [56]	0.043	0.060	-0.002	0.052
	Grad.-Input [55]	0.105	0.051	0.023	0.030
	Integ.-Grad. [60]	**0.137**	**0.112**	0.130	**0.134**
	GradCAM++ [52]	0.089	0.083	-0.001	0.063
	VarGrad [52]	0.054	0.099	**0.279**	0.093
Black-box	RISE [48]	0.020	0.074	-0.025	**0.042**
	Sobol [12]	**0.095**	**0.108**	-0.036	0.013
	WCAM (ours)	0.016	-0.037	**0.020**	-0.016

A.3 Fourier class activation map

Fourier transform The Fourier transform consists in decomposing an input signal into a sum of sinusoids, thus giving information about its frequency content. For images, the discrete Fourier transform (DFT) is mostly used. Given an image $X \in \mathbb{R}^{n \times m}$, the DFT is given by:

$$\mathcal{F}(X)(u,v) = \sum_{k=1}^{n} \sum_{l=1}^{m} X(k,l) e^{\left\{-2i\pi\left(\frac{uk}{n} + \frac{vl}{m}\right)\right\}} \tag{11}$$

Where $\mathcal{F}(X)(u,v) \in \mathbb{C}^{n \times m}$ expresses the phase and amplitude of the frequency components (u,v). Figure 11 presents an example of the Fourier amplitude and phase spectra. By convention, the amplitude spectrum is centered: lowest frequencies appear in the middle of the plot. The lighter, the higher the *energy* for the corresponding frequency. The amplitude spectrum is represented in the Nyquist square. On horizontal and vertial axes, frequencies range from $-f_e/2$ and $f_e/2$ where f_e is the *cutting frequency* of the image.

The Fourier transform is invertible and its inverse can be computed as:

$$X(k,l) = \frac{1}{nm} \sum_{u=0}^{n-1} \sum_{v=0}^{m-1} \mathcal{F}(X)(u,v) e^{\left\{2i\pi\left(\frac{ku}{n} + \frac{lv}{m}\right)\right\}} \tag{12}$$

The phase of the Fourier transform gives information on the structure of the image and the amplitude shows the distribution of the frequency components in the image.

Fourier class activation maps are applied to the amplitude spectrum of the image. We include three types of perturbations: the grid perturbation, similar to [6], the square perturbation, analogous to

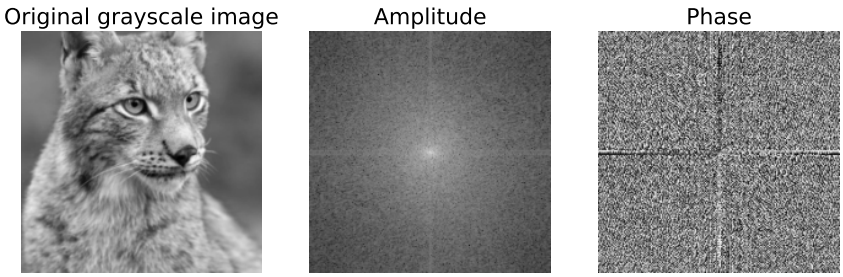


Figure 11: Fourier transform of a gray scale image.

[71] and circular perturbations. We include grid and square perturbation for comparison purposes. However, applying these perturbations to the spectrum may induce ripples during reconstruction. On the other hand, circular masks are "cleaner" and should generate less ripples. Therefore, we consider this perturbation as the one that should be adopted by default.

Fourier attribution method parameterization A crucial choice for the estimation of the importance of the frequency components is the perturbation applied to the frequency spectrum. Earlier works such as [6] apply grid masks to the frequency spectrum, whereas [71] use squared masks. We raise awareness on the fact that during image reconstruction, such masks may induce reconstruction artifacts and that it can be hard to distinguish between the effect of the reconstruction artifact such as ripples (or Gibbs phenomenon) and the perturbation on the model's outputs. To minimize the occurrence of such artifacts, we introduce a new class of circular masks that are applied to the magnitude spectrum, in all directions. Figure Figure 12 depicts examples of attribution using the FAM with grid, square and circular masks.

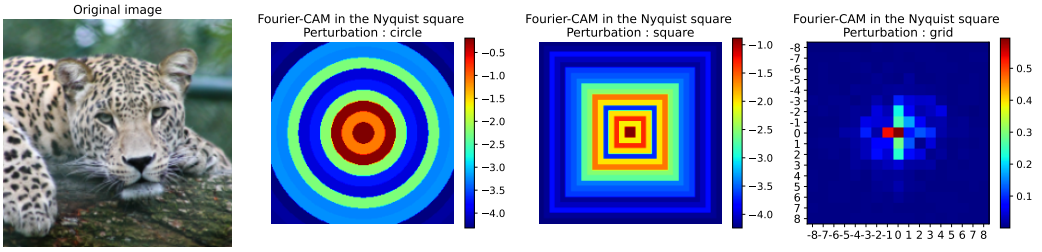
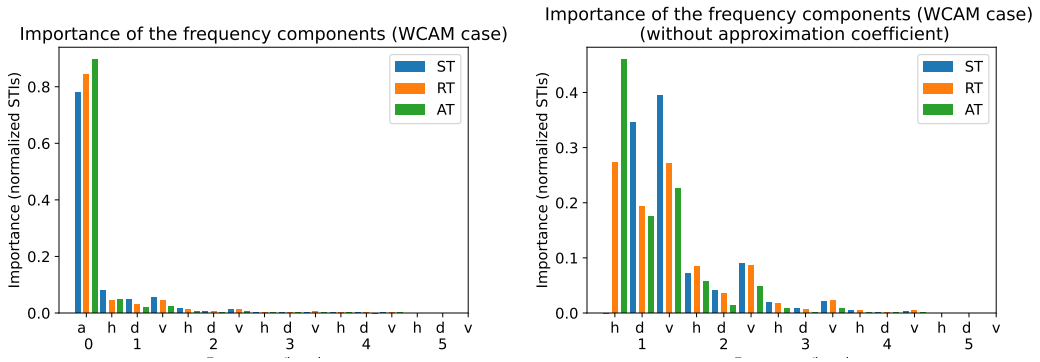


Figure 12: Examples of Fourier-CAM represented in the Nyquist space. They are generated from three different perturbations of the frequency spectrum : circle, square and grid.

Fourier and wavelet attribution quantify frequency importance similarly When considering wavelets, one may preferably consider scales instead of frequencies. Roughly speaking, we can consider that the finer scales correspond to the higher frequencies. Therefore, if the WCAM and the Fourier attribution method behave the same, the WCAM should highlight the prevalence of the coarser scales (*i.e.*, lowest frequencies) in a model's prediction. On Figure 13, one can see that the importance of the coarser scales is indeed more important *and* that the baseline training method puts more weights on the finer scales than the robust and adversarial training methods.

FAM: linking attribution and robustness We introduce the Fourier attribution method to integrate existing works on robustness into our setting. The FAM only allow to visualize the importance of frequencies in the prediction, with no information on the spatial localization. On Figure 13 we show that WCAM and FAM yield similar results regarding the quantification of the importance of the



(a) Representation with all coefficients

(b) Representation with detail coefficients only

Figure 13: Representation of the scales in the WCAM as frequencies. Levels (numbered 1 to 5) are ranked by descending order, from the coarser scales (*i.e.*, lowest frequencies) to the finest scales (*i.e.* highest frequencies). For a given level, labels "d", "h" "v" stand for the "diagonal", "horizontal" and "vertical" components. Computation of the importance for a given component is based on the sum of the Sobol indices for this components normalized by the total number of indices devoted to this scale. Computations are based on a 5-level dyadic decomposition of the input. "ST", "RT" and "AT" stand for "standard training", "robust training" and "adversarial training" respectively.

different frequency components to a model's prediction. In appendix B.1 we show that we can recover existing results from earlier works in the literature with our Fourier attribution method. However, we argue that the WCAM should be preferred, as they include a quantification of the importance of frequencies (through scales) and localization to a model's decision.

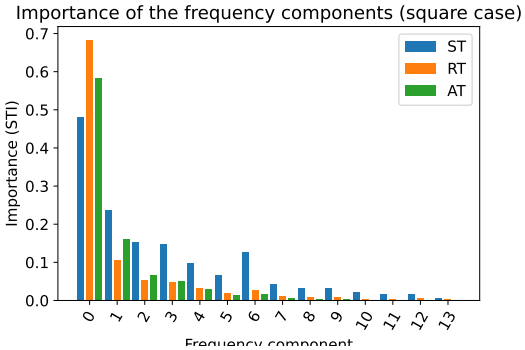
B Additional results

B.1 Equivalence between FAM and existing works

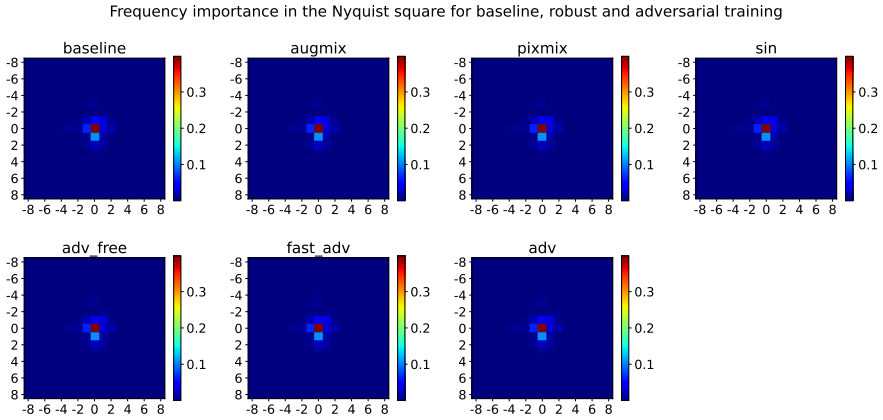
In this section, we show that our method is consistent with existing works that decomposed the magnitude spectrum of input images to recover the importance of the different frequency components. With the FAM, we included a parameterization to compute the masks: circular (our baseline method), square and grid. The square and grid perturbations come from [71] and [6] respectively. These works used a similar method based on a perturbation of the magnitude spectrum and Shapeley values to recover the importance of the frequency components. [71] applied their method to benchmark standard and robust methods while [6] focused on baseline and adversarial training methods. Their main results can be summarized as:

1. Lower frequencies have the highest importance for the prediction,
2. Robust and adversarial training methods yield a higher importance of the lowest frequencies in the prediction than the baseline training method.

Figure 14 shows that we recover the same results with our method than previous works. Besides, computing the frequency importance with our method is faster than with the methods of earlier works as the computational cost is lower for computing the of the Sobol indices than the Shapeley values.



(a) Frequency importance based on the squared perturbation [71]



(b) Frequency importance based on the grid perturbation [6]

Figure 14: Frequency importance in the prediction for robust (RT), standard (ST) and adversarial (AT) training methods. To compute the importance, we use either square masks (Figure 14a) or a grid (Figure 14b) to perturb the magnitude spectrum of the input image.

B.2 Alternative datasets

B.2.1 ImageNet Renditions

This dataset, introduced by [24], aims at benchmarking model robustness to shifts in the texture of objects from ImageNet. This dataset dubbed ImageNet-R(enditions) contains cartoon, sketch, paintings from objects. This dataset is intented in highlighting and mitigating the texture bias of ImageNet-trained models [19]. Figure 15, Figure 16 and Figure 17 present some examples of model evaluation on ImageNet-C. We plot the image and the spatial WCAM (upper row) and the WCAM (lower row). Results are reproducible with the notebook `imagenet-r.ipynb`.

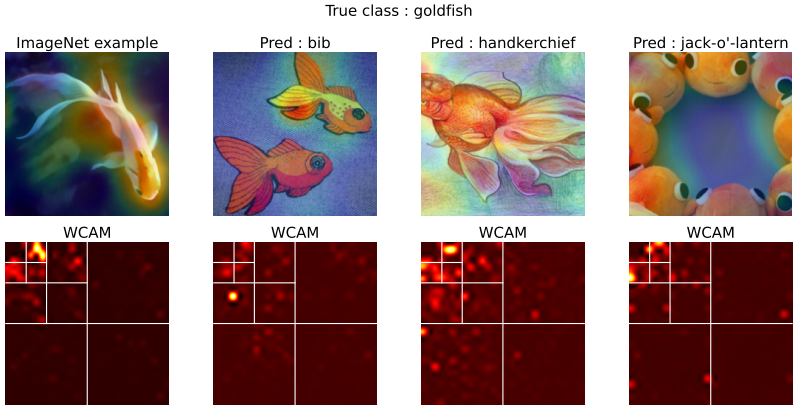


Figure 15: Predictions from ImageNet (left column) and from ImageNet-R (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see that for the prediction of the bib, the model mainly relied on details at the 2-4 pixel scale, which are white and may have been confused with the strides of the clothes.

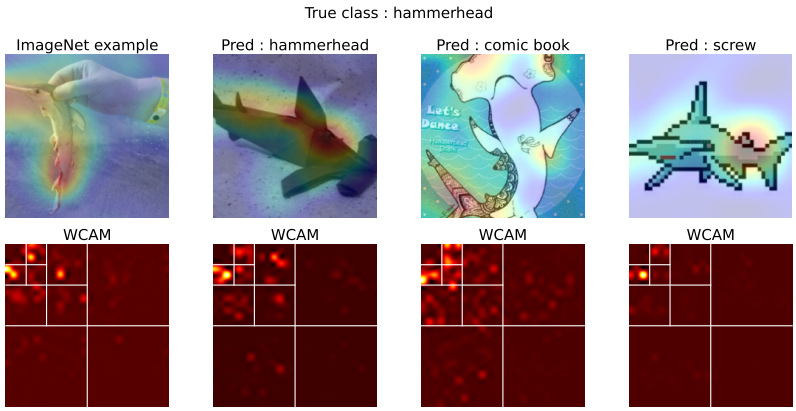


Figure 16: Predictions from ImageNet (left column) and from ImageNet-R (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see that for the comic book prediction, the model was disrupted by details at small scales (1-2 pixel) around the edges of the hammerhead. Finally for the screw prediction, the pixellisation mislead the model. We can see that the overall shape has little importance in the final prediction. The model is looking for textures.

B.2.2 Natural adversarial examples (ImageNet-A)

[27] introduced this dataset containing "natural" adversarial examples to highlight the reliance of deep models on learning shortcuts [17]. For instance, models can rely on the background to predict the foreground class, e.g. pasture to predict a cow. ImageNet-A(dversarial) contains hard samples obtained by adversarial filtration. This corresponds to clean samples coming from the original ImageNet dataset for which the model (a ResNet-50) made a wrong prediction. Figure 18, Figure 19 and Figure 20 present some examples of model evaluation on ImageNet-A. We plot the image and the spatial WCAM (upper row) and the WCAM (lower row). Results are reproducible with the notebook `imagenet-a.ipynb`.

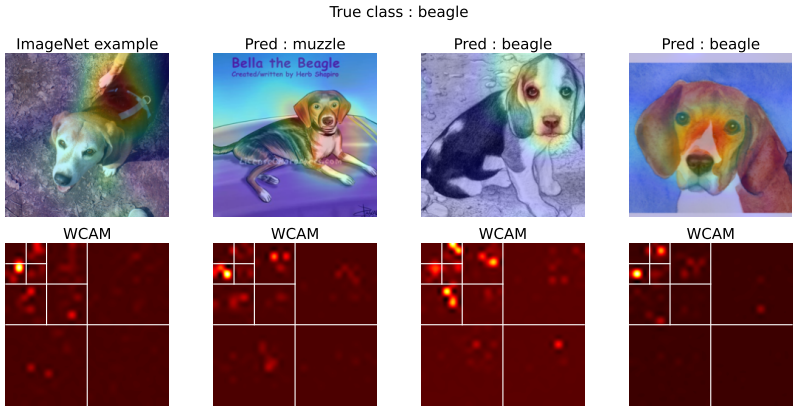


Figure 17: Predictions from ImageNet (left column) and from ImageNet-R (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see that the consistent prediction is probably caused by the small details (1-2 pixel scale) that are still present on the two rightmost images. On the other hand, as soon as these details disappear, the shape of the muzzle is not sufficient to predict the class of the dog. Again, the shape is not decisive in the prediction.

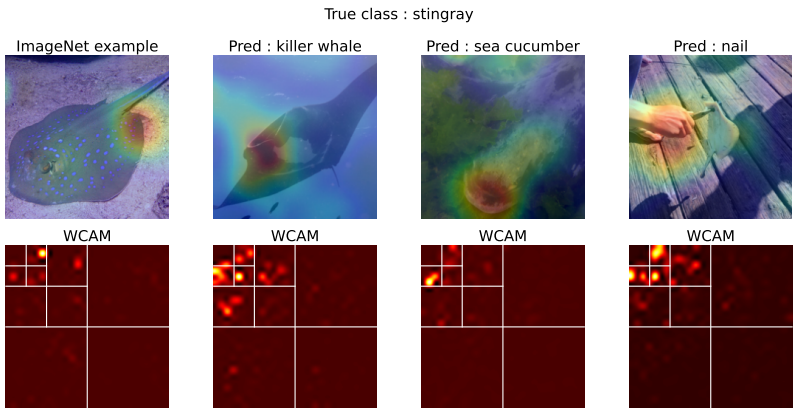


Figure 18: Predictions from ImageNet (left column) and from ImageNet-A (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see that for all but the last image (rightmost column) the model focuses on the correct area (spatially). However, when it predicts a killer whale, we can see that it considers finer scales (2-4 pixel) than for the prediction of a stingray (leftmost column).

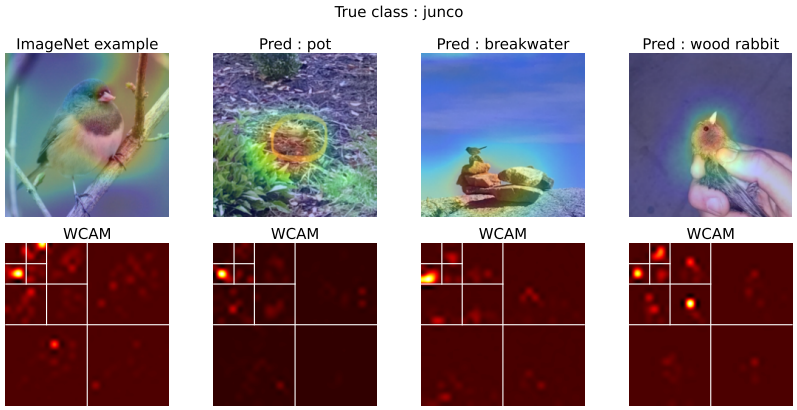


Figure 19: Predictions from ImageNet (left column) and from ImageNet-A (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see on the third column (prediction "breakwater") that although the model looks at the correct location, it does not see the bird at all. For the rightmost column, it likely confuses the bird's beak with rabbit's ears: it sees them in the 2-4 pixels scale whereas for the baseline example, the beak appeared at the pixel scale

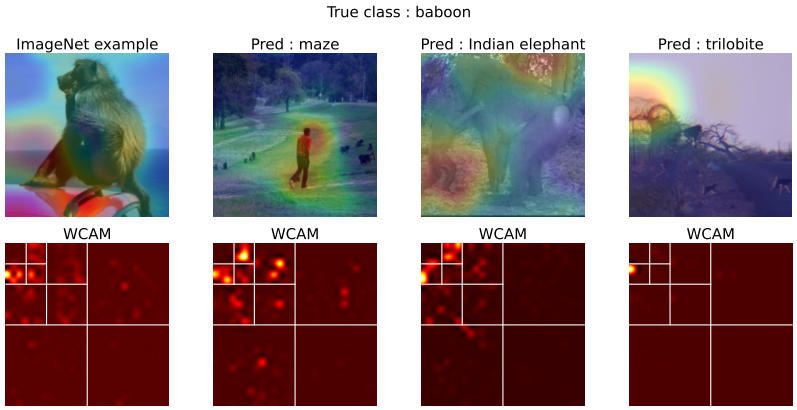


Figure 20: Predictions from ImageNet (left column) and from ImageNet-A (three remaining columns). The upper row plots the image and the spatial WCAM, the bottom row the WCAM. We can see on the second plot (prediction "maze"), that the model does not see the baboons at all. We can see that the model focuses on similar scales but predicts another class: the wrong prediction may be caused by the disruptions at the 1-pixel to 2-4 pixel scale.

B.3 Reconstruction based on the most important Wavelet coefficients

Figure 21 illustrates the reconstruction from the most important wavelet coefficients only. Wavelet coefficients are ranked using their associated Sobol indices. We reconstruct images with a gradually increasing number of coefficients.

As our grid size is parameterized on 28, we have at most $28 \times 28 = 784$ coefficients. On the first column of Figure 21, we plot the original image. In the middle, we plot the image reconstructed using only the most important Wavelet coefficient. On the rightmost column, we reconstruct the image using only the positive coefficients. We also indicate the number of coefficients it corresponds to.

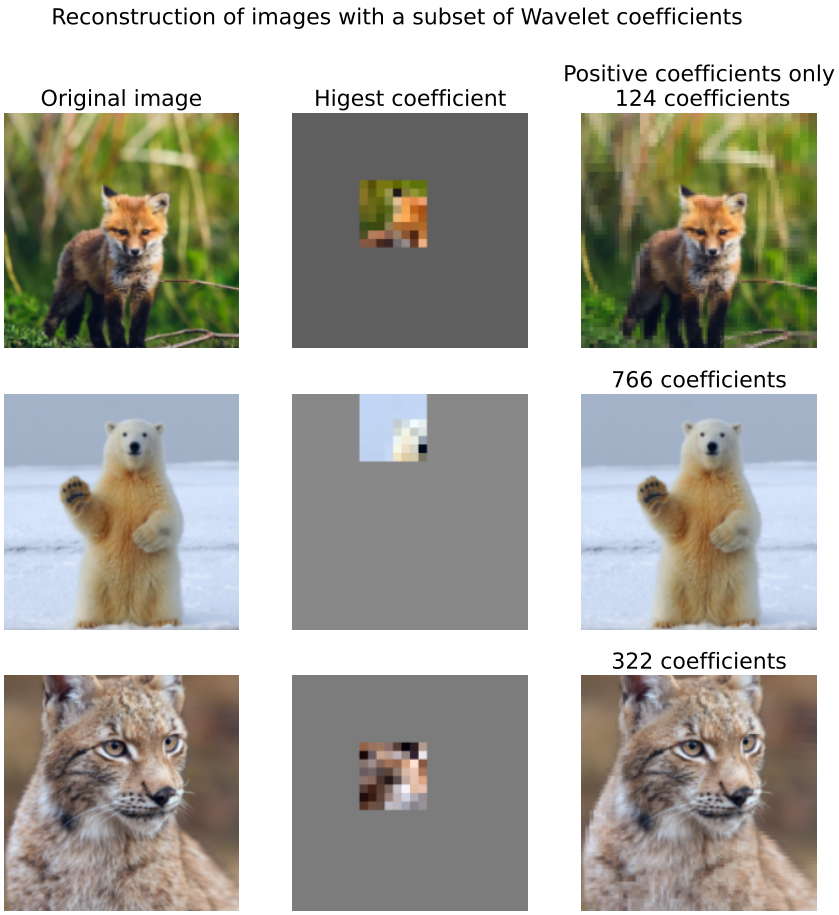


Figure 21: Example of reconstruction using the most important Wavelet coefficients. On the column in the center, we reconstruct the image with only the most important coefficient. On the rightmost column, we plot the image reconstructed with only the n positive coefficients.

Reconstruction depth and robustness Reconstruction depth correlates with the robustness of the prediction. As it can be seen from Figure 22, a relationship seems to emerge between the robustness to corruption and the reconstruction depth. This result is constant across model backbones and training approaches.

We can notice that "robust" methods have the lowest reconstruction depth on average, followed by adversarial methods and finally baselines. Surprisingly, ViT and ResNets behave similarly. This could highlight the primary role of augmentations for learning robust representations.

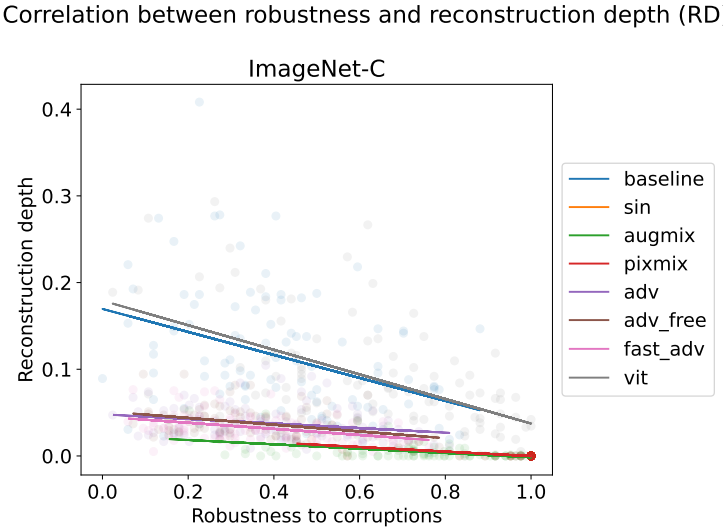


Figure 22: Correlation between the robustness to image corruptions and the reconstruction depth. The robustness to corruption is the ratio of corrupted (resp. edited) images that are correctly predicted by the model. We observe that through all model backbones and training methods, the lower the reconstruction depth, the higher the robustness. This result holds for both image corruptions (ImageNet-C [25], leftmost plot) and image editing (ImageNet-E [41], rightmost plot).

B.4 Zoom consistency: experimental details

B.4.1 Evaluation of the zoom consistency

To evaluate the zoom consistency of the models, we randomly sample 100 images from the ImageNet validation set. For these 100 images, we apply the standard resize and crop transforms (`transforms.Resize(256)`) and (`transforms.CenterCrop(224)`) to generate the "regular" images. To generate the zoomed-in images, we use the following transform: (`transforms.Resize(512)`) and (`transforms.CenterCrop(224)`). As depicted on the upper row of Figure 23, the zoomed-in image has the same resolution (224×224) but is zoomed twice on the center of the image. In the zoomed-in image, our transform preserves the scales. Therefore, the level at the 2-4 pixel scale corresponds to the level at the 1-2 pixel scale on the regular image.

Equivalence between the levels in the regular and zoomed-in image To evaluate the importance of the levels, we compute the WCAM on the regular and zoomed-in image. As depicted on Figure 23, for the zoomed-in image we add a fourth level. This level corresponds to the 1-2 pixel scale, which was previously invisible on the image. If the model relies on coefficients at these levels, it means that it uses previously invisible information to make its prediction. On the other hand, the distribution of importances should remain relatively stable between the levels if the model leverages information at the same scale between the two images.

Quantification of the importance of the levels To compute the importance of the levels, we sum the importance at each level (for the diagonal, horizontal and vertical sections). We then normalize the importance vector.

B.4.2 Full results table

Table 4 and Table 5 are the disaggregated version of Table 1. Table 4 reports the coefficients for the vanilla ResNet and the "robust methods", Table 5 reports the coefficients for the adversarial methods and the ViT.

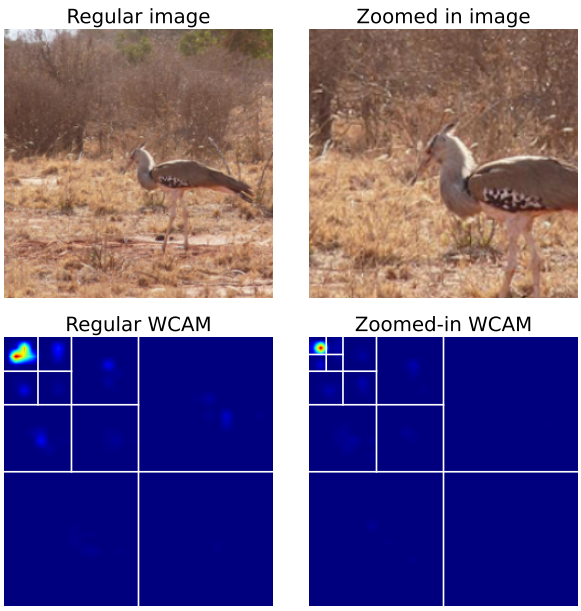


Figure 23: Computation of the WCAM for a prediction made by the vanilla ResNet on a regular (left) and zoomed-in twice (right) image. The WCAM is computed for four levels on the zoomed-in image. The three last levels correspond to the same scales as on the regular image.

Table 4: **Importance** (quantified by the Sobol indices of the WCAM) of each scale level in the prediction for regular **(Reg.)** and zoomed-in ($\times 2$) transforms of 100 images sampled from ImageNet. When we zoom in on the image, we decompose it into four levels. The highest level (4) contains information at scales that can only be seen on the zoomed-in image. The bolded value indicates the share of the *new* information leveraged by the model to classify the image. Standard errors in parenthesis.

Level	*Baseline*		*Augmix* [26]		*PixMix* [28]		*SIN* [20]	
	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$
0	0.837 (0.064)	0.752 (0.137)	0.892 (0.088)	0.873 (0.088)	0.904 (0.074)	0.863 (0.114)	0.890 (0.082)	0.873 (0.102)
1	0.130 (0.053)	0.190 (0.107)	0.080 (0.070)	0.093 (0.070)	0.069 (0.055)	0.100 (0.086)	0.082 (0.063)	0.091 (0.075)
2	0.028 (0.012)	0.047 (0.030)	0.024 (0.023)	0.029 (0.030)	0.022 (0.023)	0.032 (0.036)	0.024 (0.025)	0.030 (0.032)
3	0.005 (0.002)	0.010 (0.006)	0.004 (0.005)	0.005 (0.005)	0.004 (0.006)	0.005 (0.005)	0.004 (0.006)	0.005 (0.004)
4	(-) (-)	**0.001** (0.001)	(-) (-)	**0.001** (0.001)	(-) (-)	**0.001** (0.001)	(-) (-)	**0.001** (0.001)

B.5 Additional visualizations

B.5.1 Baseline model

In this section we plot additional examples similar to figure 4, with different images, for the vanilla Resnet and the Vision transformer. These images we generated automatically following the notebook `1-spectral-decomposition.ipynb` accessible on the project's repository. The interested reader can generate similar plots for any model backbone.

Table 5: **Importance** (quantified by the Sobol indices of the WCAM) of each scale level in the prediction for regular **(Reg.)** and zoomed-in ($\times 2$) transforms of 100 images sampled from ImageNet. When we zoom in on the image, we decompose it into four levels. The highest level (4) contains information at scales that can only be seen on the zoomed-in image. The bolded value indicates the share of the *new* information leveraged by the model to classify the image. Standard errors in parenthesis.

Level	*Adv* [42]		*Adv-free* [53]		*Adv-fast* [64]		*ViT* [11]	
	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$	Reg.	$\times 2$
0	0.953	0.903	0.951	0.906	0.959	0.909	0.830	0.611
	(0.036)	(0.077)	(0.033)	(0.081)	(0.030)	(0.081)	(0.075)	(0.164)
1	0.040	0.082	0.042	0.079	0.035	0.078	0.137	0.295
	(0.031)	(0.069)	(0.029)	(0.073)	(0.027)	(0.072)	(0.063)	(0.132)
2	0.006	0.013	0.006	0.013	0.005	0.012	0.029	0.078
	(0.006)	(0.010)	(0.006)	(0.011)	(0.005)	(0.011)	(0.018)	(0.043)
3	0.001	0.001	0.001	0.002	0.001	0.001	0.004	0.014
	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)	(0.001)	(0.002)	(0.008)
4	(-)	**0.000**	(-)	**0.000**	(-)	**0.000**	(-)	**0.002**
	(-)	(0.000)	(-)	(0.000)	(-)	(0.000)	(-)	(0.001)

Vanilla ResNet Figure 24, Figure 25, Figure 26 present additional examples for the ResNet architecture.

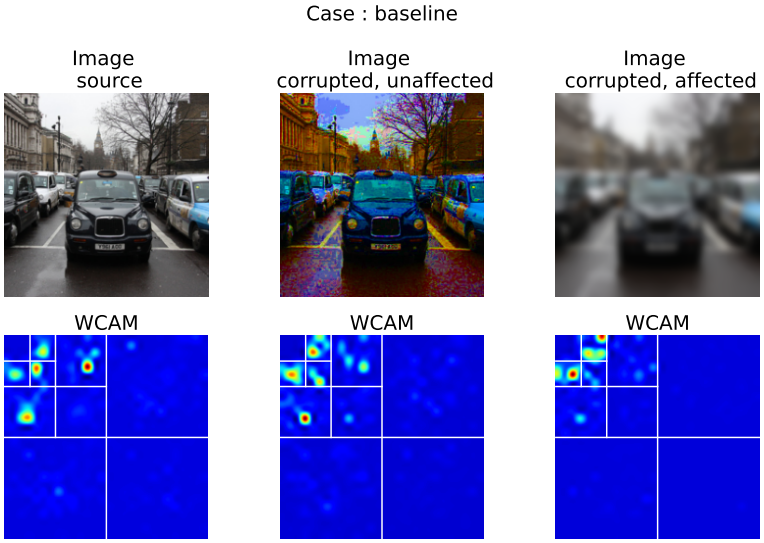


Figure 24: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

Vision transformer Figure 27, Figure 28, Figure 29 present additional examples for the Vision transformers architecture.

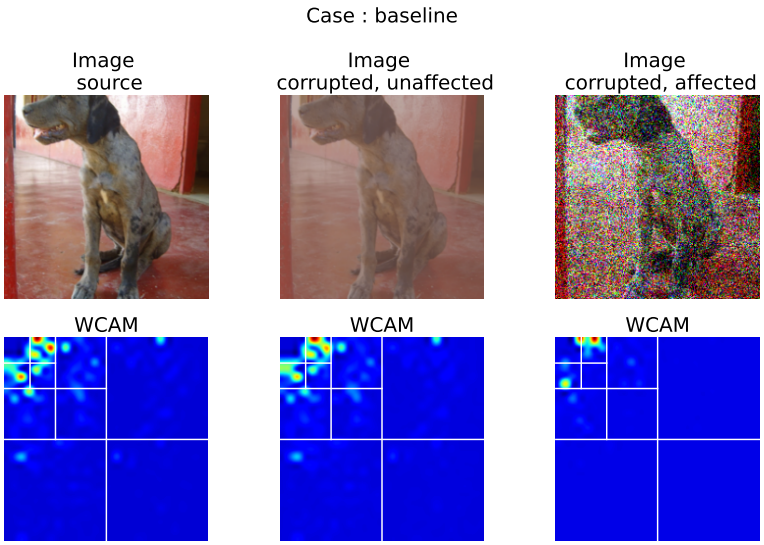


Figure 25: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

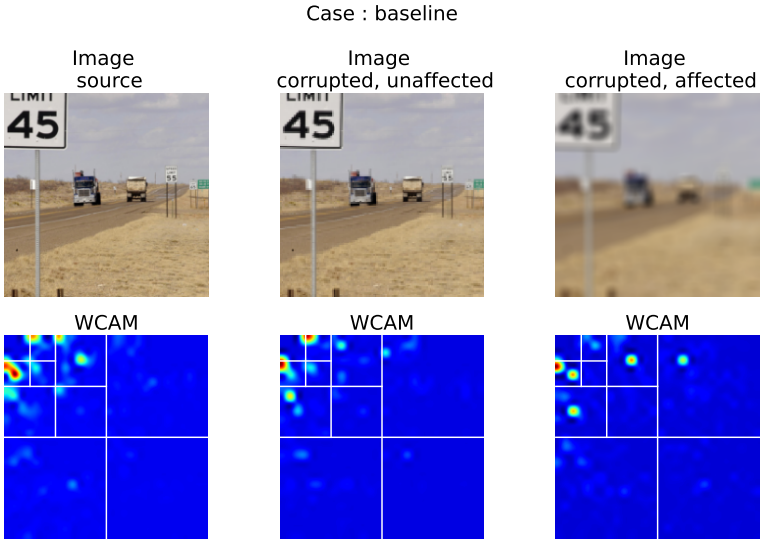


Figure 26: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

B.5.2 Alternative models

On Figure 30, Figure 31, Figure 32, we plot the WCAM for a clean and corrupted image across various model baselines. On this section's plots, each column represents a model instance. The first row presents the WCAM for the clean image and the bottom row for the corrupted image, which has been mispredicted. All code to replicate these figures is accessible on the project's repository.

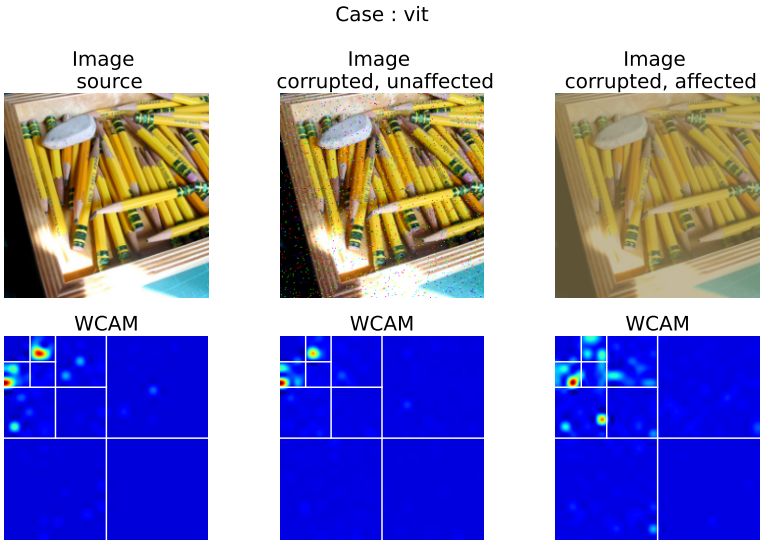


Figure 27: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

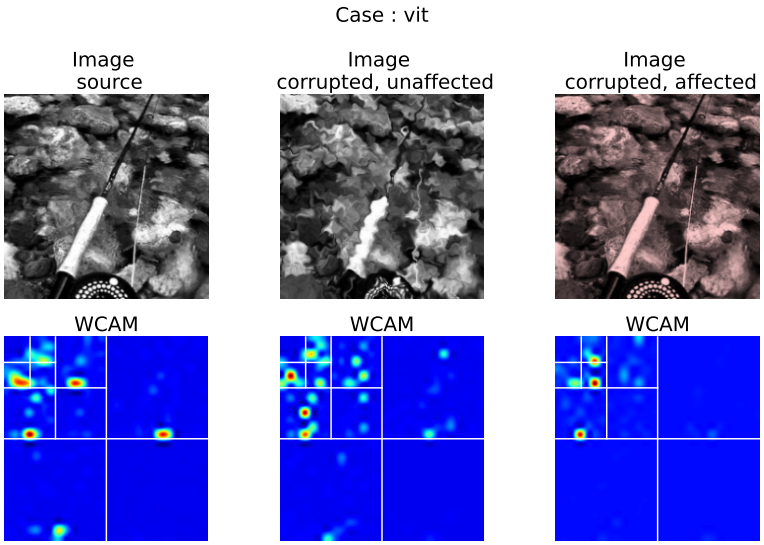


Figure 28: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

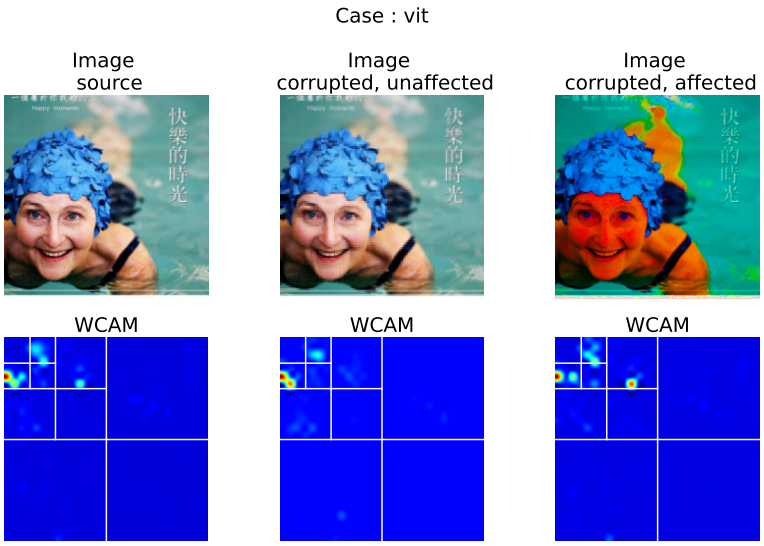


Figure 29: Additional visualizations of the effect of a corruption on a model's prediction through the lenses of the WCAM

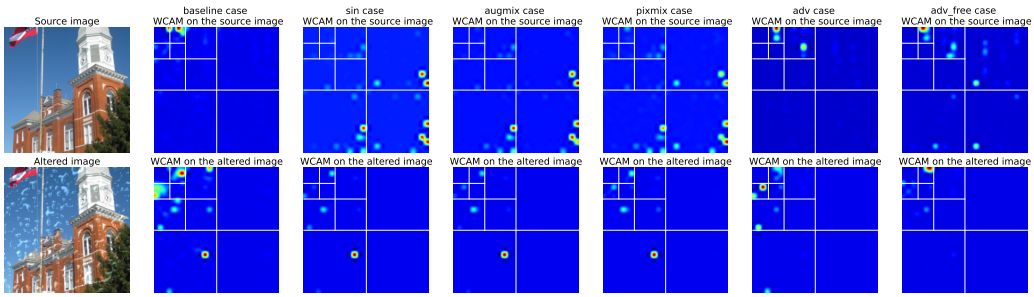


Figure 30: Additional visualization of the WCAM for a clean (upper row) and corrupted sample (bottom row). Each column but the leftmost represents the WCAM of a training method.

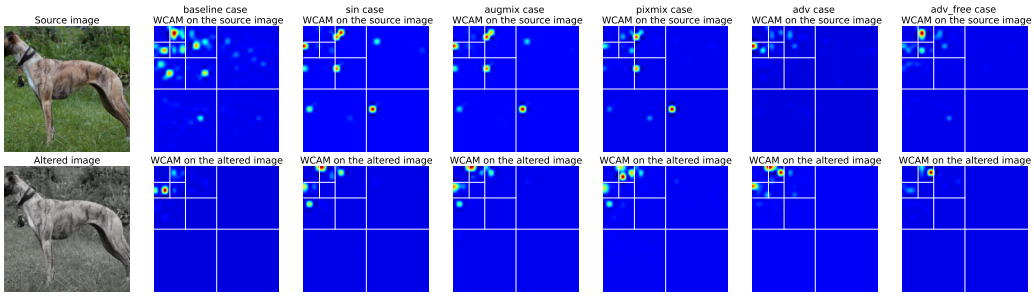


Figure 31: Additional visualization of the WCAM for a clean (upper row) and corrupted sample (bottom row). Each column but the leftmost represents the WCAM of a training method.

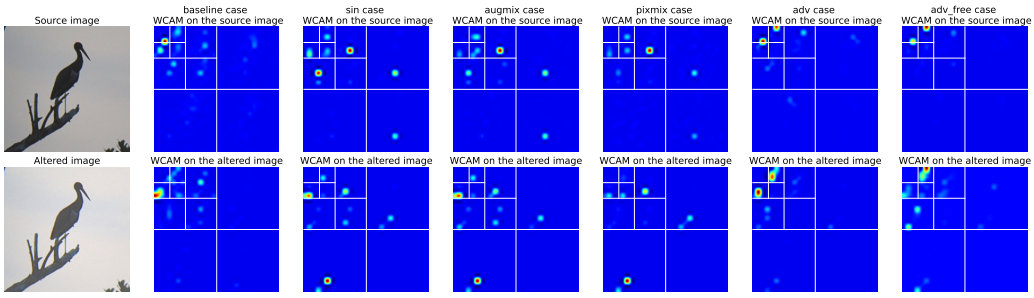


Figure 32: Additional visualization of the WCAM for a clean (upper row) and corrupted sample (bottom row). Each column but the leftmost represents the WCAM of a training method.