A Implementational details

A.1 Effect of smoothing and integration on the explanations

Figure 6 illustrates the improvement of the quality of the explanations by smoothing (third row) or integrating (fourth row) the gradients, compared to their raw values (second row). Smoothing follows Equation 2 and integration follows Equation 3.

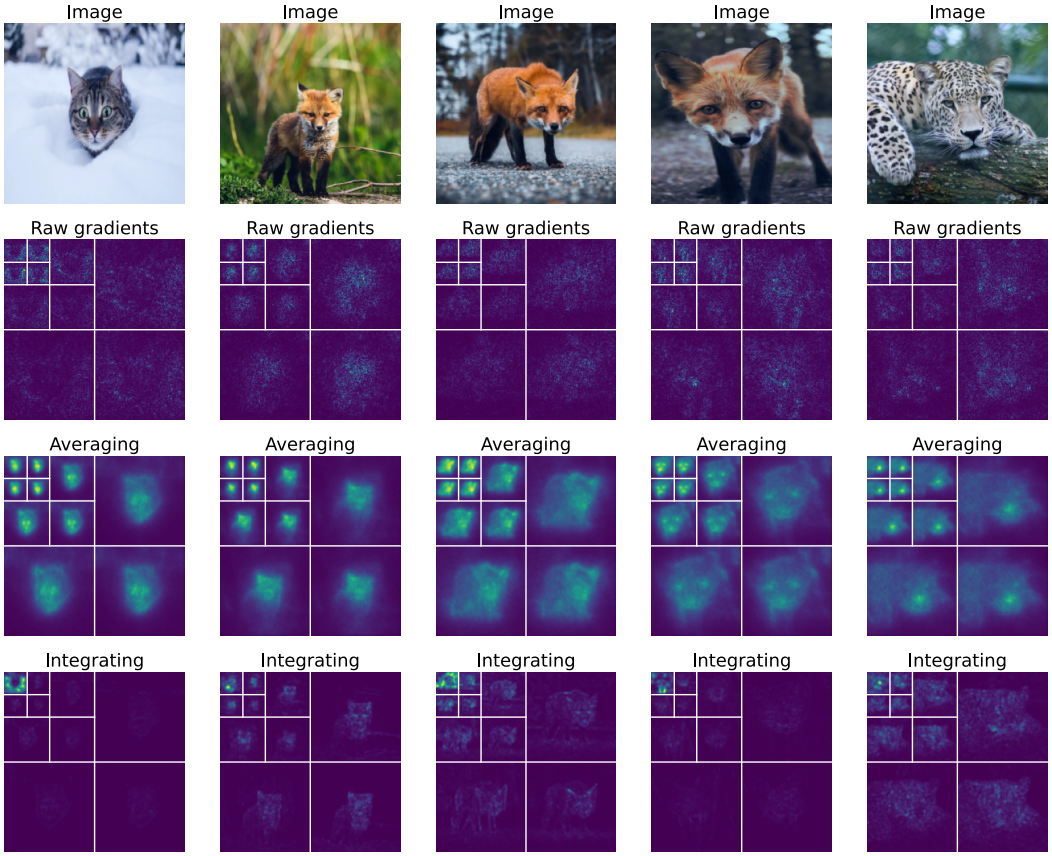

Figure 6: Effect of smoothing (3rd row) and integration (4th row) compared to the raw gradients (2nd) raw when computing the WAM of the images (1st row). The explanations are depicted in the wavelet domain.

We can see that both methods display complementary properties regarding the explanation. WAM_{SG} enables to visualize highlights the important locations within scales, while WAM_{IG} emphasizes on see the relative importance of each scale.

A.2 Benchmark construction

All benchmarks reported in this work were carried out on a server running Ubuntu 20.04.6 and on a single NVIDIA TITAN Xp GPU with 12 GB of VRAM with CUDA 12.5. The data to replicate the experiments can be downloaded in the repository accessible at this URL `https://doi.org/10.5281/zenodo.13873810`, and the source code is accessible from the Git repository.

Images. Our models' parameterizations for benchmarking WAM on images are the following:

- ResNet: we consider the `resnet18` variant,
- EfficientNet: we consider the `tf_efficientnet_b0.ns_jft_in1k` variant,
- ConvNext: we consider the `convnext_small.fb_in22k_ft_in1k_384` variant,

- DeiT: we consider the `deit_tiny_patch16_224.fb_in1k`

All models are retrieved from the PyTorch Image Models (Wightman, 2019) repository. We load the model with the pre-trained weights and directly evaluate them on the validation set of ImageNet. We implement the SmoothGrad, GradCAM, and GradCAM Plus Plus methods ourselves and use the Captum library (Kokhlikyan et al., 2020) for implementing the Intergrated Gradients and the Saliency methods.

Audio. For audio, we use the same technical infrastructure to evaluate our method. We use the CNN classification model of Kumar et al. (2018) as our black-box model to explain. We consider a single model as alternative models (Huang & Leanos, 2018; Wilkinghoff, 2021; Lopez-Meyer et al., 2021) are only variations around the same topology. We add 0 dB white noise to the ESC-50 samples using the pseudocode displayed in Figure 7.

```
# Input: audio (input audio signal, array of int16)
# Output: noisy_audio (audio with added Gaussian noise, array of int16)

def add_gaussian_noise(audio):
  # Convert the audio to float32 for safe computation
  audio_float = convert_to_float32(audio)

  # Calculate RMS (Root Mean Square) of the audio signal
  rms_signal = sqrt(mean(audio_float ** 2))

  # Generate Gaussian noise
  noise = random_normal_distribution(mean=0,
                                     std=1,
                                     shape=audio_float.shape)

  # Calculate RMS of the generated noise
  rms_noise = sqrt(mean(noise ** 2))

  # Scale noise to have the same RMS as the audio signal
  noise = noise * (rms_signal / rms_noise)

  # Add noise to the audio signal
  noisy_audio_float = audio_float + noise

  # Clip the noisy audio to ensure it stays within the int16 range
  noisy_audio_clipped = clip(noisy_audio_float, -32768, 32767)

  # Convert the clipped noisy audio back to int16
  noisy_audio = convert_to_int16(noisy_audio_clipped)

  return noisy_audio
```

Figure 7: Pseudo-code for adding Gaussian noise to audio

B Complements on the quantitative evaluation

B.1 Insertion and deletion

Insertion and deletion are two evaluation metrics proposed by Petsiuk et al. (2018). These metrics are "area-under-curve" (AUC) metrics, which report the change of in the predicted probability for the image class when inserting (resp. removing) meaningful information highlighted by the attribution method. Petsiuk et al. (2018) initially defined this metric for images, where the important features

correspond to pixels. We expand this metric to wavelet coefficients, thus enabling a computation of the Insertion and the Deletion for any modality.

Both metrics consider an input in a baseline state. Insertion consists in adding the most important features identified by the attribution method. Formally, at step k with a subset u_k of important features (which correspond in our case in wavelet coefficients) at step k,

$$\text{Insertion}^{(k)} = \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{-u_k}=\boldsymbol{x}_0]}), \tag{5}$$

where $\boldsymbol{f}(.)$ is the predicted probability and $-u$ denotes the complementary set of u. We add features by decreasing order of importance and for $k_1 \leq k_2, u_{k_1} \subseteq u_{k_2}$, i.e., we gradually add more and more features until we eventually recover the full input $\boldsymbol{x}$.

The deletion performs the opposite operation where we start from a plain input with all variables and we gradually set features in the baseline state $\boldsymbol{x}_0$, from the most important to the less important. We have

$$\text{Deletion}^{(k)} = \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{u_k}=\boldsymbol{x}_0]}). \tag{6}$$

Finally, for the insertion and the deletion, we measure the AUC, which is comprised between 0 and 1. Given K steps, the Insertion score of the feature attribution $\boldsymbol{\gamma}$ for the model $\boldsymbol{f}$ is

$$\text{Ins}(\boldsymbol{f}, \boldsymbol{\gamma}) = \sum_{k=1}^{K} \text{Insertion}^{(k)} \Delta_k = \sum_{k=1}^{K} \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{-u_k}=\boldsymbol{x}_0]}) \Delta_k, \tag{7}$$

where Δ_k is the width between two subintervals. The computation is analogous for $\text{Del}(\boldsymbol{f}, \boldsymbol{\gamma})$.

If the attribution method picks relevant features, then only including them (resp. only removing them) should result in a large increase (resp. large decrease) in the predicted probability. Therefore, the AUC should be close to 1 for the insertion and close to 0 for the deletion. We set the baseline to $\boldsymbol{x}_0 = 0$.

B.2 Complementary results

B.2.1 Definitions

μ-Fidelity. The μ-Fidelity is a correlation metric. It measures the correlation between the decrease of the predicted probabilities when features are in a baseline state and the importance of these features. We have

$$\mu\text{-Fidelity} = \underset{\substack{u \subseteq \{1,\dots,K\}, \\ |u|=d}}{\text{Corr}} \left(\sum_{i \in u} \boldsymbol{g}(\boldsymbol{x}_i), \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{x}_u=\boldsymbol{x}_0}) \right), \tag{8}$$

where $\boldsymbol{g}$ is the explanation function (i.e., the explanation method), which quantifies the importance of the set of features u.

Faithfulness on Spectra. The Faithfulness on Spectra **(FF**, Parekh et al., 2022**)** measures how important is the generated interpretation for a classifier. The metric is calculated by measuring the drop in class-specific logit value $\boldsymbol{f}(\boldsymbol{x})_c$, when the masked out portion of the interpretation mask $\boldsymbol{m}_{\boldsymbol{\gamma}}$ is input to the classifier. This amounts to calculating,

$$\text{FF}_{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})_c - \boldsymbol{f}\left(\boldsymbol{x} \odot (\boldsymbol{1} - \boldsymbol{m}_{\boldsymbol{\gamma}})\right)_c. \tag{9}$$

It should be noted that this strategy to simulate removal may introduce artifacts in the input that can affect the classifier's output unpredictably. Also, interpretations on samples with poor fidelity can lead to negative $\text{FF}_{\boldsymbol{x}}$. These observations point to this metric's potential instability and outlying values. Thus, we report the final faithfulness of the system as the median of $\text{FF}_{\boldsymbol{x}}$ over the test set, denoted by FF. A positive FF would signify that interpretations are faithful to the classifier.

Input Fidelity. The Input Fidelity (**Fid-In**, Paissan et al., 2023) measures if the classifier outputs the same class prediction on the masked-in portion of the input image. It is defined as,

$$\text{Fid-In} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[\arg\max_c \boldsymbol{f}(\boldsymbol{x}_i)_c = \arg\max_c \boldsymbol{f}_c\left(\boldsymbol{x}_i \odot \boldsymbol{m}_{\boldsymbol{\gamma}}\right)\right], \tag{10}$$

where $\mathbb{I}$ denotes the indicator function and, again, larger values are better.

B.2.2 Results

Images. Table 3 reports the results using the μ-Fidelity (Bhatt et al., 2021) compared to other methods. We can see that the performance measured by the μ-Fidelity is more in line between our method and the existing approaches.

Table 3: **μ-Fidelity** (Bhatt et al., 2021) score obtained on 1,000 images from the validation set of ImageNet and for different model architectures. Higher is better. **Bolded** results are the best and underlined values are the second best.

Method	*ResNet-18*	*ConvNext*	*EfficientNet*	*DeiT*	Mean
Saliency	0.154	0.186	0.180	0.195	0.179
Integrated Gradients	**0.228**	0.223	**0.219**	**0.241**	**0.228**
GradCAM	0.141	0.216	0.149	0.151	0.164
GradCAM ++	0.135	0.212	0.141	0.222	0.178
SmoothGrad	0.220	0.227	0.211	0.230	0.222
Guided Backpropagation	0.216	0.229	**0.234**	0.226	0.226
WAM$_{SG}$ (ours)	0.215	0.208	0.213	0.216	0.213
WAM$_{IG}$ (ours)	0.170	0.166	0.165	0.182	0.171

Table 4 reports the Insertion and Deletion (Petsiuk et al., 2018) scores compared to other attribution methods. We can see that the good results reported in Table 1 are mostly driven by our method's very good results in Insertion. We can see that WAM systematically outperforms the other methods in both Insertion and Deletion.

Audio. Table 5 displays the results of the quantitative evaluation of WAM when the metrics are computed from the mel-spectrogram. We can see that we still achieve state-of-the-art performance for the Faithfulness of spectra and are in line with other metrics for the Input Fidelity. These results highlight the added value of computing the metrics from the wavelet coefficients rather than from the mel-spectrogram.

B.3 Randomization check

Movitation. The sanity checks introduced by Adebayo et al. (2018) aim at assessing whether an explanation depends on the model's parameters and the input labels. These tests aim to assess the faithfulness of an explanation beyond visual evaluation. The randomization test evaluates whether an explanation depends on the model's parameters. Parameters have a strong effect on a model's performance. Therefore, for a saliency method to be useful for debugging or analyzing a model, it should be sensitive to its parameters. Adebayo et al. (2018) proposed different methods to randomize the model parameters. One particularly interesting implementation is the "cascading" randomization, in which the weights are randomized from the top to the bottom layers.

Method and results. We compute WAM for our 1,000 ImageNet validation samples for a set of increasingly randomized models. A randomized layer is a layer that we reset at its initial value. We consider a ResNet-18 and randomize its layers from the shallowest `conv1` to the deepest `fc`. We then compute the rank correlation (or Pearson correlation coefficient, Pearson, 1896) between the

Table 4: **Insertion** and **Deletion** (Petsiuk et al., 2018) scores obtained on 1,000 images from the validation set of ImageNet and for different model architectures. For insertion, higher is better, and for deletion, lower is better. **Bolded** results are the best and underlined values are the second best.

	Method	*ResNet*	*ConvNext*	*EfficientNet*	*DeiT*	Mean
Insertion (↑)	Saliency	0.134	0.381	0.148	0.194	0.214
	Integrated Gradients	0.087	0.305	0.113	0.095	0.150
	GradCAM	0.413	0.495	0.364	0.352	0.406
	GradCAM ++	0.452	0.562	0.350	0.313	0.419
	SmoothGrad	0.106	0.253	0.129	0.108	0.149
	Guided Backpropagation	0.090	0.332	0.117	0.093	0.158
	WAM_{SG} (ours)	**0.557**	**0.606**	**0.447**	**0.546**	**0.539**
	WAM_{IG} (ours)	0.422	0.557	0.419	0.492	0.473
Deletion (↓)	Saliency	0.109	0.349	0.140	0.156	0.189
	Integrated Gradients	0.087	0.304	0.113	0.092	0.149
	GradCAM	0.279	0.423	0.303	0.190	0.299
	GradCAM ++	0.268	0.507	0.300	0.269	0.336
	SmoothGrad	0.083	0.253	0.119	0.104	0.140
	Guided Backpropagation	0.089	0.331	0.116	0.092	0.157
	WAM_{SG} (ours)	0.119	0.272	0.097	0.123	0.153
	WAM_{IG} (ours)	**0.078**	**0.198**	**0.049**	**0.072**	**0.099**

Table 5: **Evaluation scores** of WAM and comparison with baselines on 400 audios from the first fold of ESC-50. The column "ESC" indicates that the samples are unaltered. The column "+WN" indicates that the samples have 0 dB Gaussian white noise. We report the results with explanations generated from the mel-spectrogram of the waveform. **Bolded** results are the best and underlined values are the second best.

Method	**Faithfulness** (↑)		**Insertion** (↑)		**Deletion**(↓)		**FF** (↑)		**Fid-In** (↑)	
	ESC50	+WN	ESC50	+WN	ESC50	+WN	ESC50	+WN	ESC50	+WN
IntegratedGradients	**0.264**	**0.310**	0.267	**0.312**	**0.047**	**0.045**	**0.207**	**0.207**	0.220	0.225
GradCAM	0.072	0.073	**0.274**	0.274	0.201	0.201	0.137	0.135	**0.517**	**0.542**
Saliency	0.066	0.065	0.220	0.221	0.154	0.156	0.166	0.168	0.253	0.245
SmoothGrad	0.184	0.184	0.251	0.251	0.067	0.067	0.193	0.194	0.177	0.175
WAM_{SG} (ours)	0.009	0.007	0.169	0.166	0.159	0.161	0.152	0.149	0.117	0.122
WAM_{IG} (ours)	0.000	0.004	0.168	0.171	0.168	0.167	0.149	0.149	0.105	0.128

WAM of the original, fully-trained model (labeled `orig`) and the randomized models (labeled by the name of the layer until which they are randomized).

Figure 8 presents the results. The dotted line represents the average rank correlation across the 1,000 images, and the intervals represent the 95% confidence intervals. We can see that the correlation between WAM s significantly decreases as the randomization increases, thereby showing that WAM is sensitive to the model's parameters. The lower decrease that we observe for WAM_{IG} compared to WAM_{SG} comes from the fact that WAM_{IG} reflects more the inter-scales distribution of the importance than WAM_{SG} does. As pointed out by Rahaman et al. (2019) and Yin et al. (2019), even random models exhibit a spectral bias, i.e., a natural tendency to favor lower frequencies over higher ones, which translates here into the fact of naturally putting more importance on coarser scales rather than finer scales, no matter the depth of the randomization.

Illustrations. Figure 9, Figure 25, Figure 26, Figure 27 display qualitative examples to further assess how WAM varies when the randomization depth of the model increases. We can see that the explanation is no longer informative when the model is randomized.

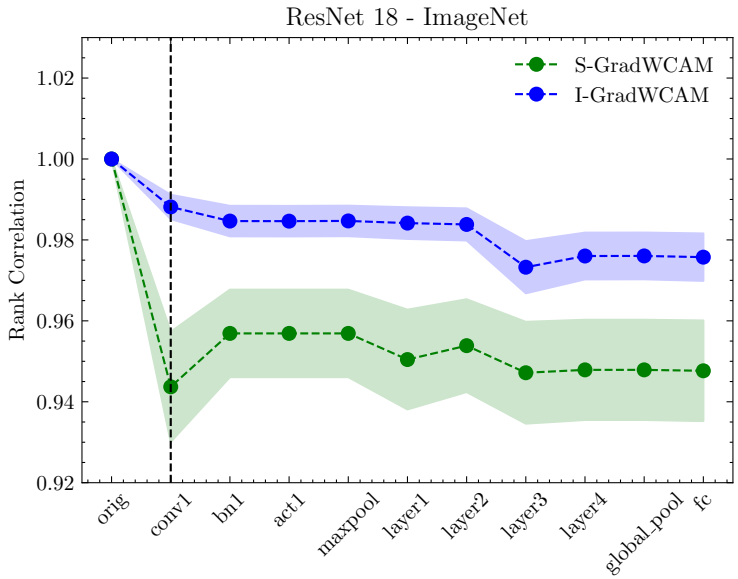

Figure 8: **Cascading randomization** of WAM for explaining a ResNet-18 on ImageNet. The y axis indicates the rank correlation between the original explanation and the explanation derived for randomization up that layer. The rank correlation is averaged over 1,000 ImageNet validation images.

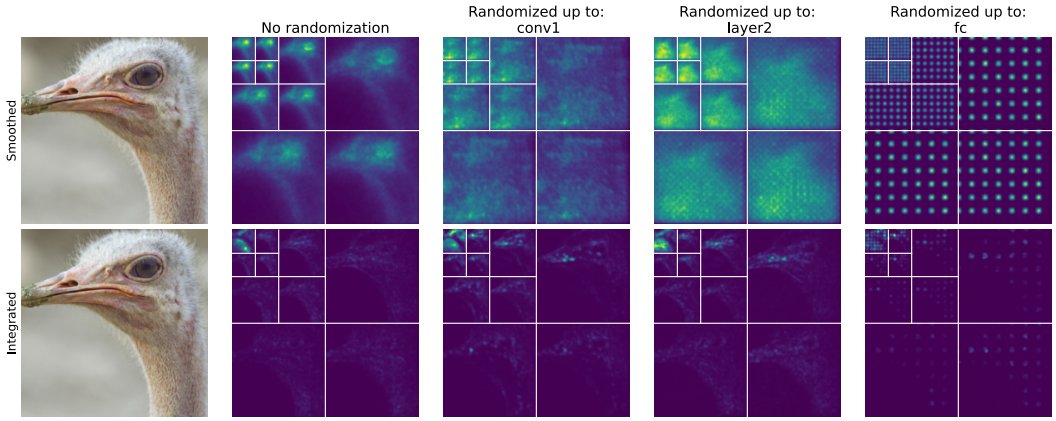

Figure 9: Illustration of the randomization test as a cascade of randomizations of the layers of the classifier. From left to right shows the explanation from WAM for an increasingly randomized ResNet-18.

C Additional Results

C.1 Revisiting Meaningful Perturbation.

In the seminal works by Fong & Vedaldi (2017) and Fong et al. (2019), a method is introduced to explain the most important parts of an image by optimizing a mask $\boldsymbol{m}$ that partially occludes certain regions. The objective is twofold: (***i***) preserve the classification score, while (***ii***) remove as much of the image content as possible to isolate the most relevant features. However, optimizing in the pixel domain presents challenges in producing smooth masks (Fong et al., 2019), necessitating various regularization techniques, smoothing operations, and data augmentations to mitigate these issues. We revisit this framework by proposing to recast the problem in the wavelet domain as a more suitable space for optimization. The wavelet domain inherently captures spatial and spec-

tral information, providing a natural structure for producing meaningful and interpretable solutions. Specifically, we solve the following optimization problem:

$$\boldsymbol{m}^\star = \underset{\boldsymbol{m}\in[0,1]^{|\mathcal{X}|}}{\arg\min}\ \boldsymbol{f}_c\left(\mathcal{W}^{-1}(\boldsymbol{z}\odot\boldsymbol{m})\right) + \alpha\|\boldsymbol{m}\|_1,$$

where $\boldsymbol{f}_c$ represents the classification score, $\mathcal{W}^{-1}$ is the inverse wavelet transform, $\boldsymbol{z}$ is the wavelet transform of the input signal, $\odot$ denotes element-wise multiplication, and α controls the sparsity of the mask $\boldsymbol{m}$. To solve this problem, we initialize the mask as $\boldsymbol{m}_0 = \mathbf{1}$ (i.e., a mask that retains all coefficients) and iteratively update it using gradient descent:

$$\boldsymbol{m}_{i+1} = \boldsymbol{m}_i - \eta\nabla_{\boldsymbol{m}_i}\left(\boldsymbol{f}_c\left(\mathcal{W}^{-1}(\boldsymbol{z}\odot\boldsymbol{m}_i)\right) + \alpha\|\boldsymbol{m}_i\|_1\right),$$

where η represents the step size and the gradient is taken with respect to $\boldsymbol{m}_i$. The optimization process continues until convergence is achieved. We call the resulting image the *minimal* image. In practice, we employ the Nadam (Dozat, 2016) optimizer, which combines the benefits of Nesterov acceleration and Adam optimization. Our approach consistently produces masks with controllable sparsity levels up to 90%, meaning that 90% of the wavelet coefficients are zeroed out, *while* maintaining a classification score comparable to or better than the original prediction. This high sparsity level suggests that the model's decision may rely on a minimal subset of wavelet coefficients.

From an interpretability perspective, our method offers significant and novel insights. Traditional meaningful perturbation methods (Fong & Vedaldi, 2017) focus on spatial localization, identifying clusters of pixels that answer the question of *where* the important features are located. However, this spatial emphasis alone provides a limited understanding of the underlying data structure. In contrast, by operating in the wavelet domain, our method captures both the *what* – the relevant scales – and the *where* – their spatial locations. This dual information enriches the explanation by revealing the location and the nature of the features influencing the model's decision. These results also show that we qualitatively recover the results from Kolek et al. (2023).

Figure 10 illustrates that minimal images derived using WAM recover the texture bias of the vanilla ResNet models trained on ImageNet, highlighted by Geirhos et al. (2019). The examples demonstrate how the model relies heavily on texture information, which is effectively isolated through our wavelet-domain optimization.

C.2 Additional results on the minimal images

Complements on "optimal" minimal images. Figure 11 illustrates the effect of the parameter α on the sparsity of the minimal images. We can see that the stronger α, the sparser the image, but at the expense of a higher logit value. We can see that the first components of the image that disappear are the background, then the colors and eventually the shape of the target class.

Static minimal images. Another method used to derive minimal images is to directly sort the gradient coefficients obtained with WAM and reconstruct the original image using decreasingly important coefficients.

We referred to the "insertion" quantile the threshold value after which the prediction of the model was correct, meaning that the partially reconstructed image contained enough information. We also considered the original image for completeness and removed gradually important coefficients. It turned out that both quantiles coincided, thus highlighting the fact that WAM identifies the necessary and sufficient amount of information for a correct prediction.

Figure 12 displays an example of a sufficient image extracted using WAM. We can see that the background information is unnecessary, contrary to details around the fox's ear and eyes. Reconstruction artifacts are caused by the fact that we independently ranked the coefficients across the three color channels.

Figure 13 presents illustrations of minimal images obtained using WAM. We can see that the background can be filtered out, while foreground details are essential. The minimal images have been obtained using information solely from WAM.

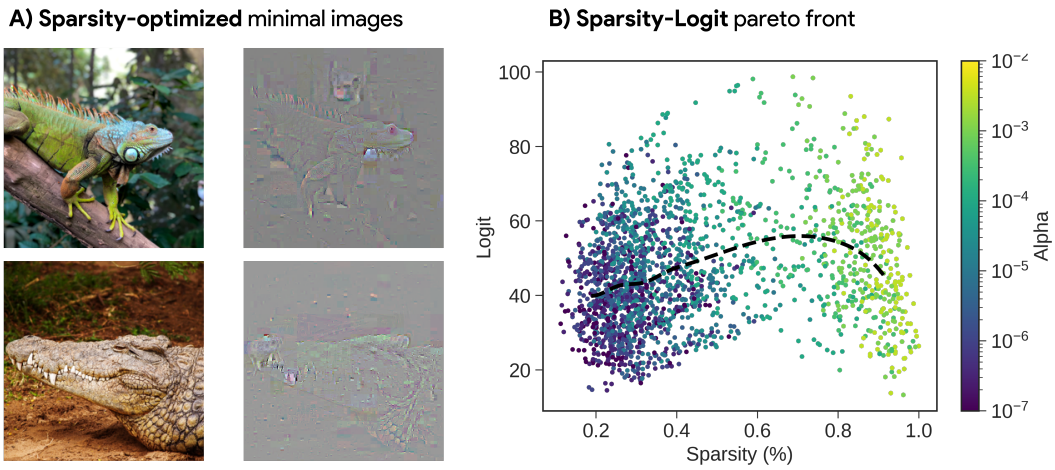

Figure 10: **A) Sparsity-optimized minimal images.** We revisit meaningful perturbation by optimizing the sparsity of the wavelet transform using masking, instead of optimizing the mask in pixel space. The displayed examples show that the resulting minimal images reveal the model's reliance on textures. **B) Sparsity Pareto front.** As α increases, the sparsity of the wavelet coefficients increases (x-axis), but beyond a certain point, too much information is lost and the logit score drops to zero. However, we observe that many components can be removed before adversely affecting the model. Results are averaged across 1,000 images optimized for 500 steps, and for α ranging in $[0, 100]$ for each image.

Figure 11: Effect of varying values of α on the sparsity of the minimal images.

C.3 Overlap experiment on audio classification

Figure 14 illustrates that WAM is able to filter relevant parts corrupted or mixed audio signals. In addition, it highlights the key part of the target signal without requiring any training. Figure 14 qualitatively illustrates application of WAM for audio signals. Herein, we perform an overlap experiment to mix a corrupting audio with a target audio to form the input audio. The model's prediction does not alter after introducing the corruption, and thus, the model is expected to still rely on parts of input audio coming from the target audio for its decision. The interpretation audio in Figure 14 generated using top wavelet coefficients provides insights into the decision process and supports this hypothesis. In particular, it almost entirely filters out corruption audio, and without requiring any training, it also clearly emphasizes key parts of target audio.

C.4 Application case: the remote sensing of photovoltaic panels

The indexation in scales of the wavelet coefficients finds a natural use case in remote sensing applications, where scales correspond to actual physical objects, as the pixels correspond to distances on the ground. Therefore, the 1-2 pixel scale (the finest details, corresponds to details that have a resolution between 20 to 40 cm on an image whose ground sampling distance is 20 cm / pixel).

On figure Figure 15, we consider a case where a classification model is trained on a source domain and deployed on a target model, thus mimicking situations where a model is used for regular updates. We can see that on the OOD images are slightly different from the source image, in the sense that they are less noisy. While one would expect the model's prediction not to change, it turns out that

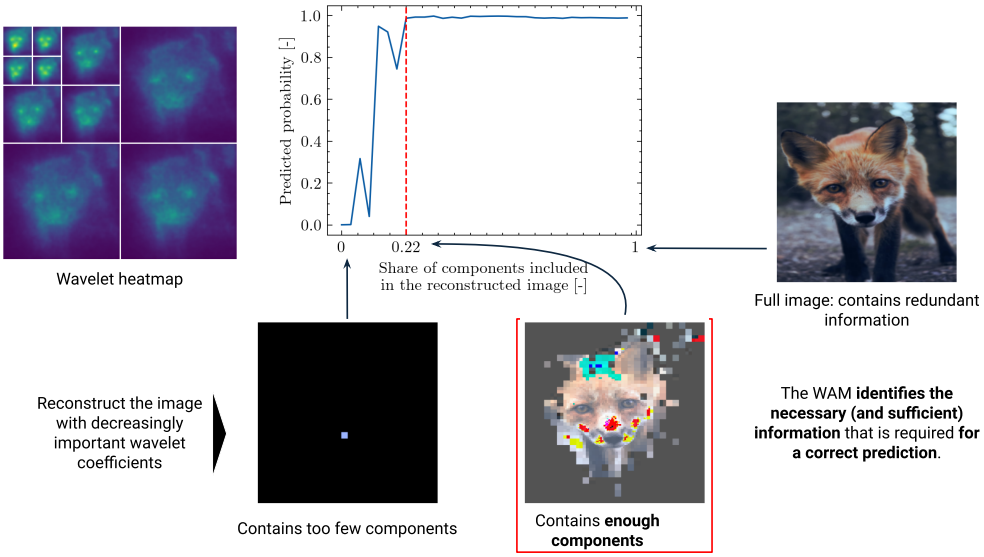

Figure 12: Extraction of the necessary and sufficient information for a correct prediction with WAM. In the example, we can see that only 22% of the components are necessary, the remaining being redundant.

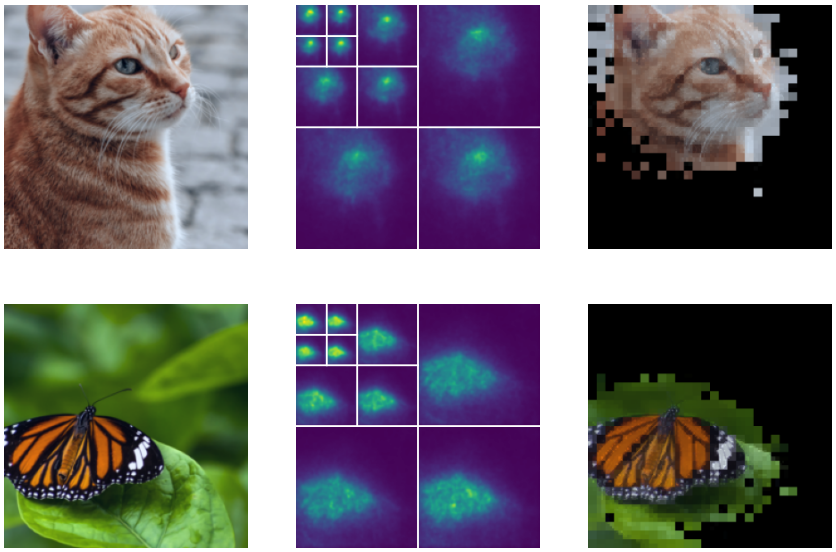

Figure 13: Example of minimal images (rightmost column), which contain enough information to correctly predict the image label. Compared to the input image (leftmost column), the background can be disregarded, but high-resolution details are necessary on the animals. The column in the center plots the wavelet heatmap obtained with WAM.

the PV panel is no longer recognized on the rightmost image. Attribution in the pixel domain only is not very informative so as to why this happens. Turning into the wavelet domain, we can better grasp why the model no longer detects the PV panel. The background noise, located mostly in the finest scales, is no longer present on the OOD image and thus the PV panel is no longer recognized.

C.5 Additional plots

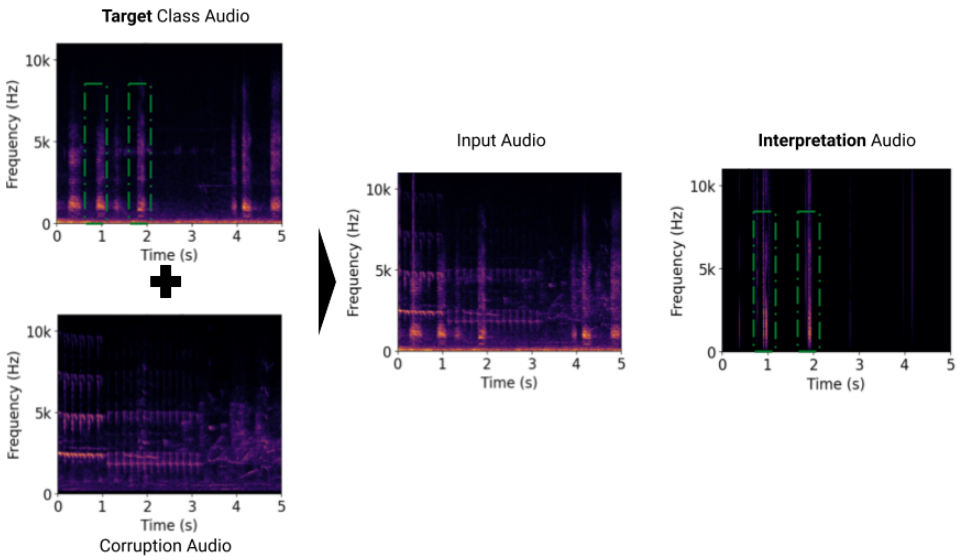

Figure 14: Qualitative illustration of WAM for audio via an Overlap experiment. The audio of the target class ('Crow') is mixed with a corrupting audio ('Chirping birds') to form the input to the classifier. Interpretation audio reconstructed with important wavelet coefficients virtually eliminate signal from the corrupting audio, and also clearly emphasize parts of the target class audio (indicated with green boxes).

Alternative visualisations of the scale decomposition brought by the WAM. Figure 16 presents an alternative visualization of the most important components at each scale. The upper row depicts the image and the image reconstructed using only the most important components, to highlight the importance of the edges at different resolutions. Figure 17 presents the wavelet decomposition of an image an the WAM visualized in the wavelet domain and back in the pixel domain to highlight to which parts of the image the structural components correspond to.

Cats and Dogs example. This example demonstrates whether the WAM can effectively distinguish between regions representing the dog and the cat in the image. In Figure 18, we observe that the WAM successfully highlights "cat-related coefficients," as the cat's nose and its corresponding regions in the wavelet domain are more active in the rightmost image. Interestingly, although less prominent, parts of the dog's face also remain significant. This suggests that while the spatial separation between the cat and dog is clear in the original image, the distinction becomes less apparent in the wavelet domain.

Additional illustrations on images. Figure 19 presents additional illustrations of WAM for images. We illustrate the important coefficients (using the IntegratedGradients variant for smoothing the gradients) in the wavelet domain (second column) as a heatmap on the original image (third column), and we show the important components on the image, i.e., what the model needs to see based on the WAM on the fourth column.

Additional minimal images. Figure 20 presents additional examples of minimal images. We can see that the color information does not appear as important for maximizing the model's prediction. On the other hand, the texture and edge information are essential. It would be interesting to replicate this method on a shape-biased model such as those proposed by Chen et al. (2020); Geirhos et al. (2019) to see whether the behavior remains the same or not.

Additional explanations on shapes. Figure 21, Figure 22, Figure 23 and Figure 24 present additional visualization on shapes.

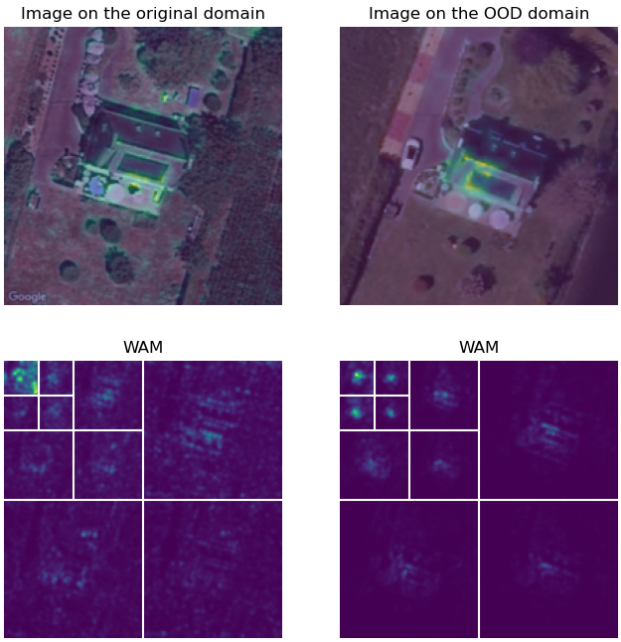

Figure 15: Application case of the WAM to remote sensing. While both images depict the same photovoltaic panel, a model no longer detects the panel when evaluated on images comping from a different distribution (OOD image, compared to images from the source distribution). Attribution only in the pixel domain is note very informative, while attribution in the wavelet domain thanks to the WAM gives hints so as to why the PV system is no longer recognized: on the image of the original domain, which is more noisy, the model relies on noise at the finest scales that is no lnoger present on the OOD image. Thus suggesting that the representation learned by the model lacks robustness.

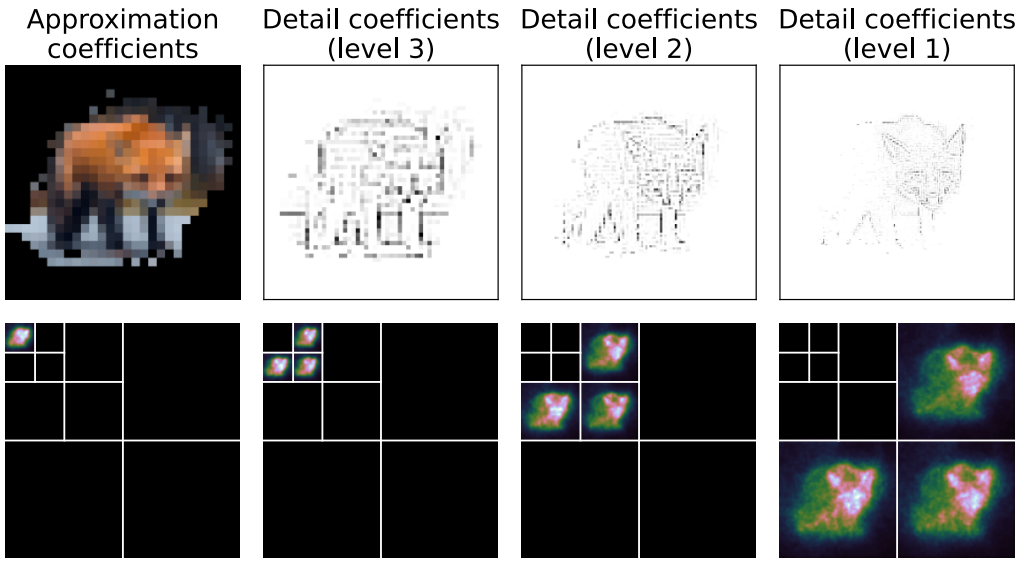

Figure 16: Illustration of the filtering of each scale with the WAM. Each column depicts the most important components of the image at each scale.

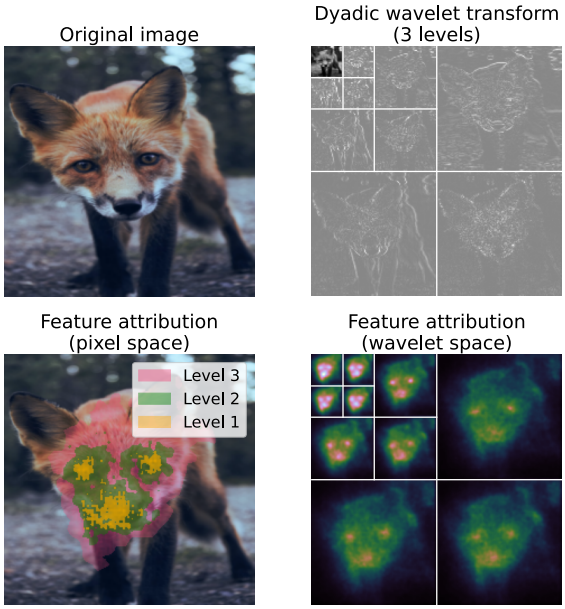

Figure 17: Decomposition of a model's prediction in the wavelet domain using the WAM. The plot on the bottom left shows in the pixel space the locatino of the important components from each level (level 1: finest details, level 2: intermediate details, level 3: coarse details).

Additional randomization checks. Figure 25, Figure 26, and Figure 27 present additional examples of randomization checks.

Additional audio overlap experiments. Figure 28 and Figure 29 present additional examples of overlap experiments on waveforms.

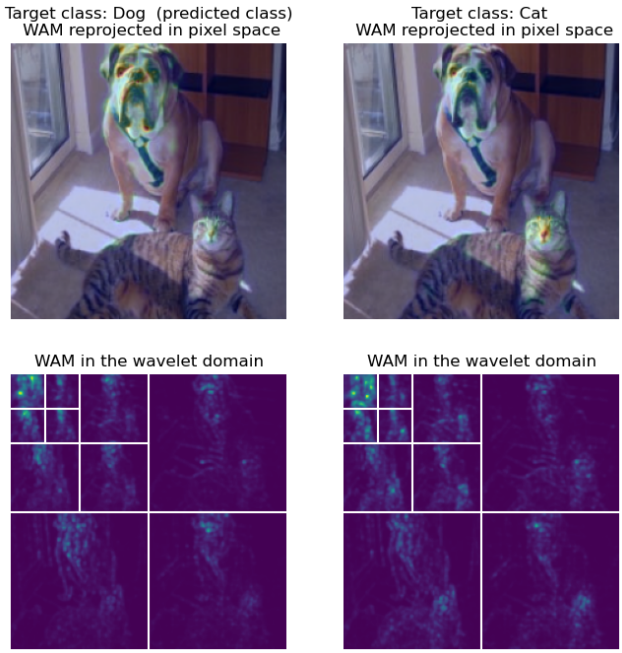

Figure 18: Cats and dogs example to highlight that the WAM points towards the regions corresponding to the target label (Dogs on the right, Cats on the left).

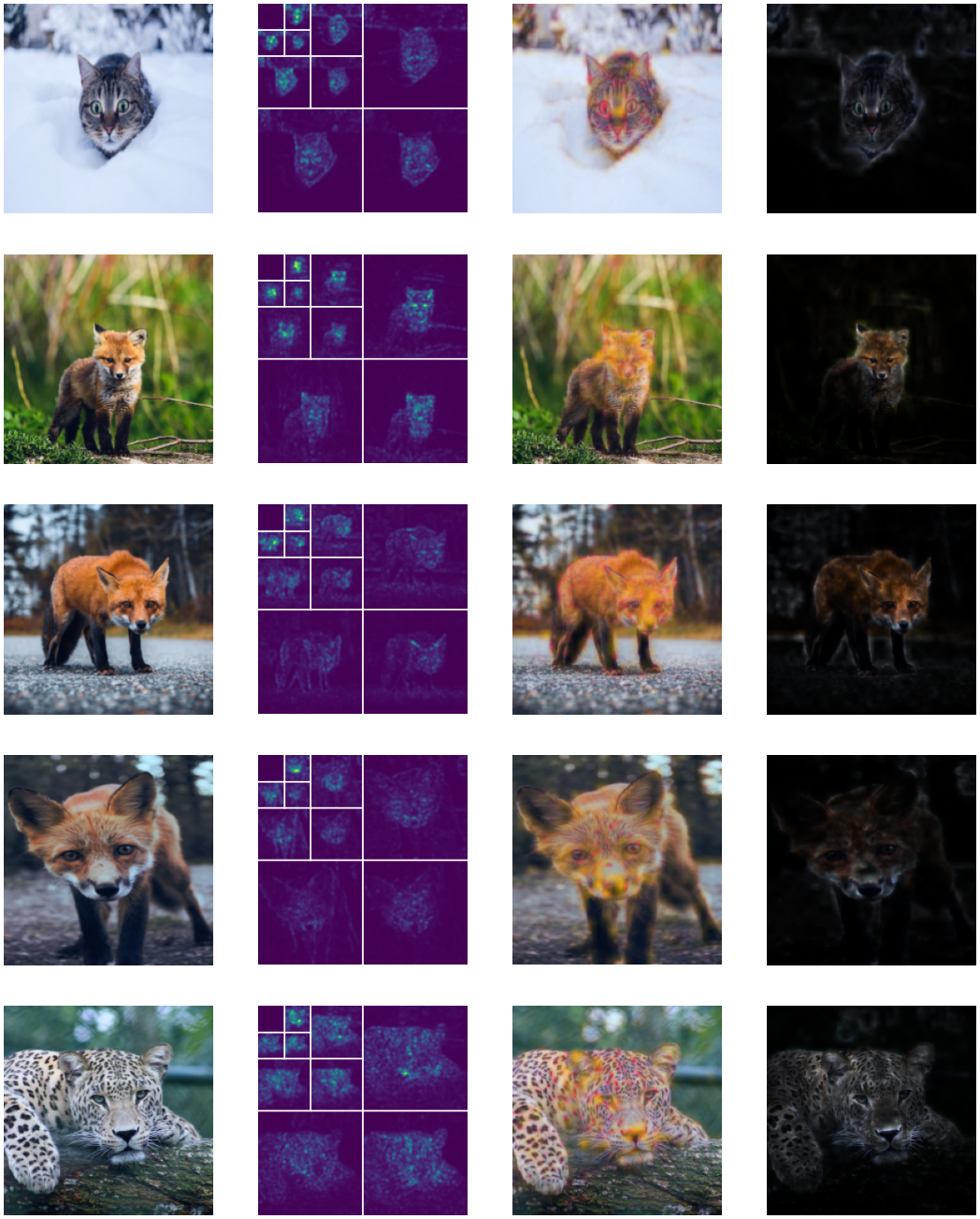

Figure 19: Additional visualizations of WAM for images. First column: original image. Second column: WAM in the wavelet domain. Third column: heatmap in the pixel domain. Fourth column: Filtered image illustrating the image regions that need to be well defined for the model to predict the image label correctly.

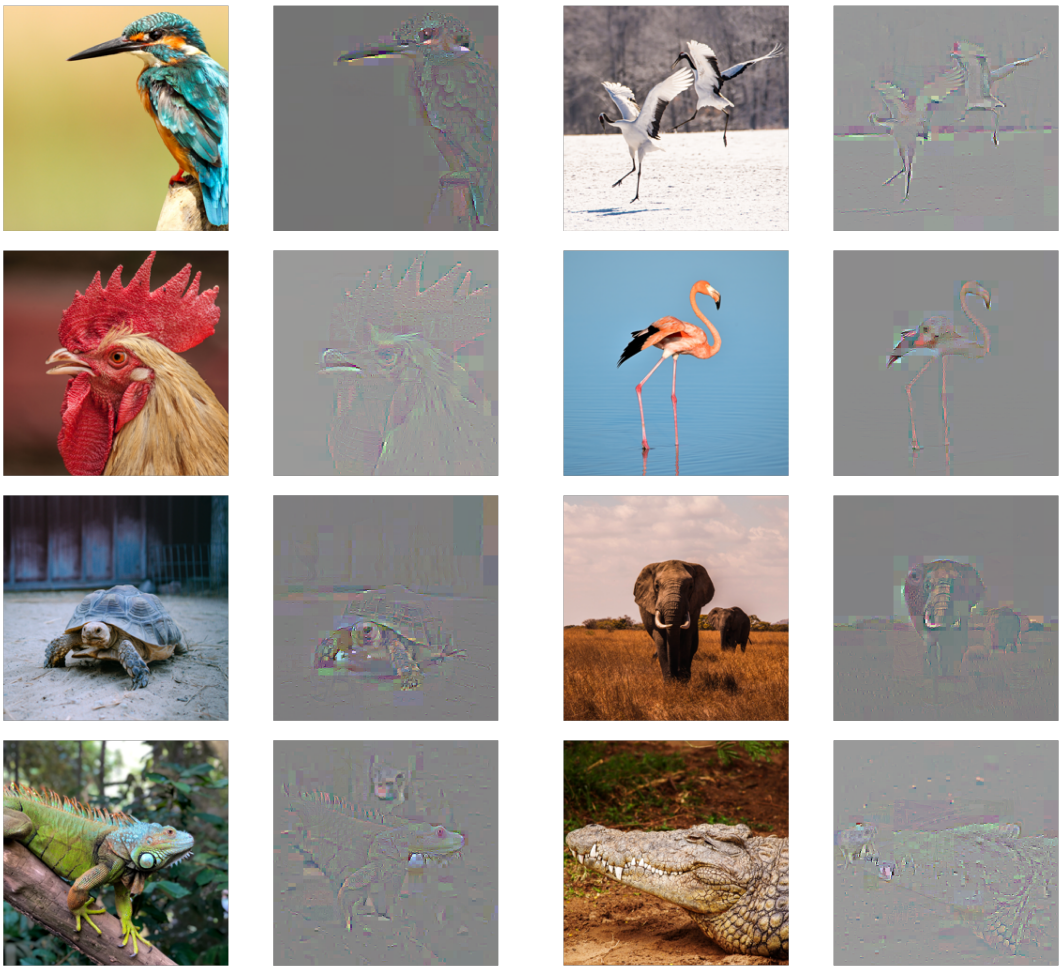

Figure 20: Additional examples of minimal images.

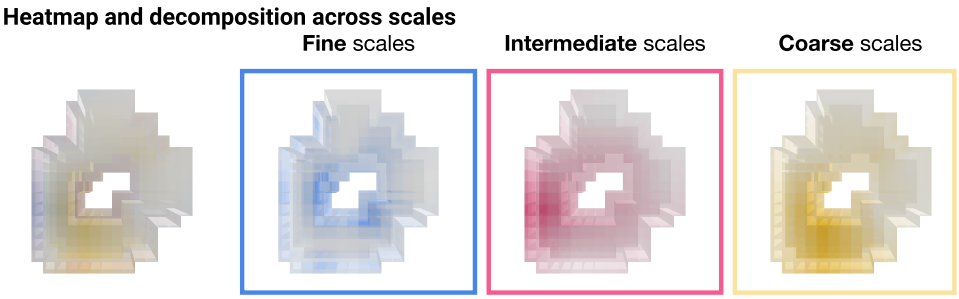

Figure 21: Decomposition of the different important scales on a voxel with WAM.

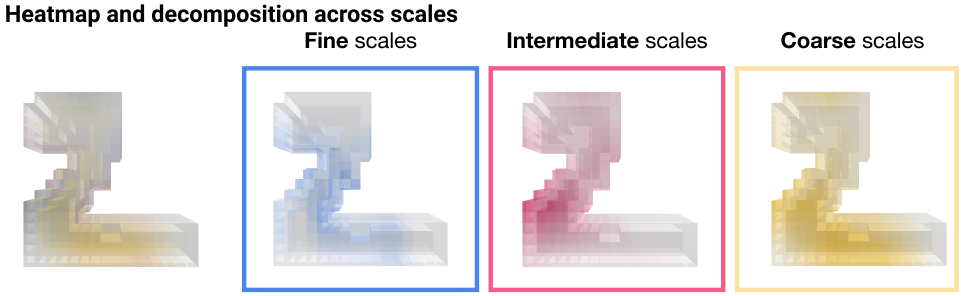

Figure 22: Decomposition of the different important scales on a voxel with WAM.

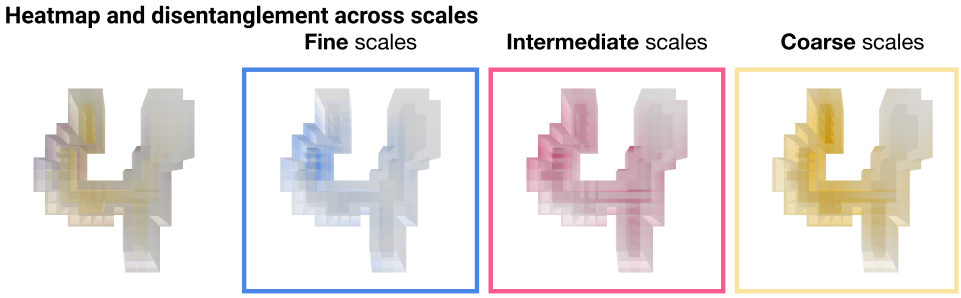

Figure 23: Decomposition of the different important scales on a voxel with WAM.

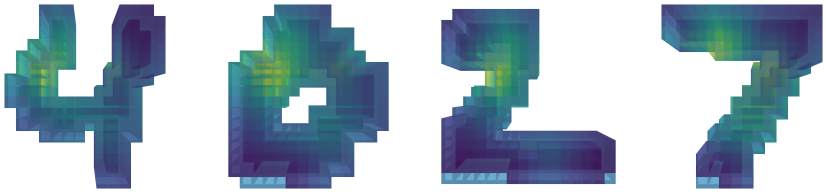

Figure 24: Heatmaps combining the importance at different scales on different voxels from 3D MNIST.

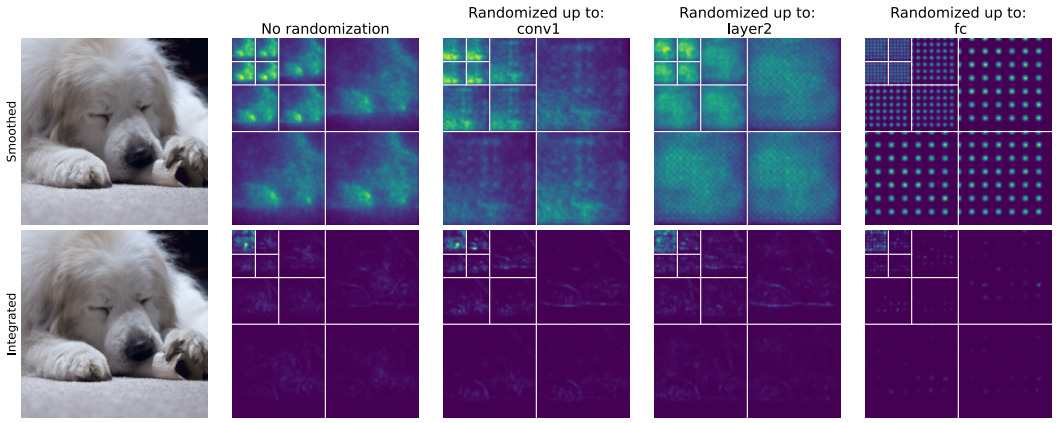

Figure 25: Illustration of the randomization test as a cascade of randomizations of the layers of the classifier. From left to right shows the explanation from the WAM for an increasingly randomized ResNet-18.

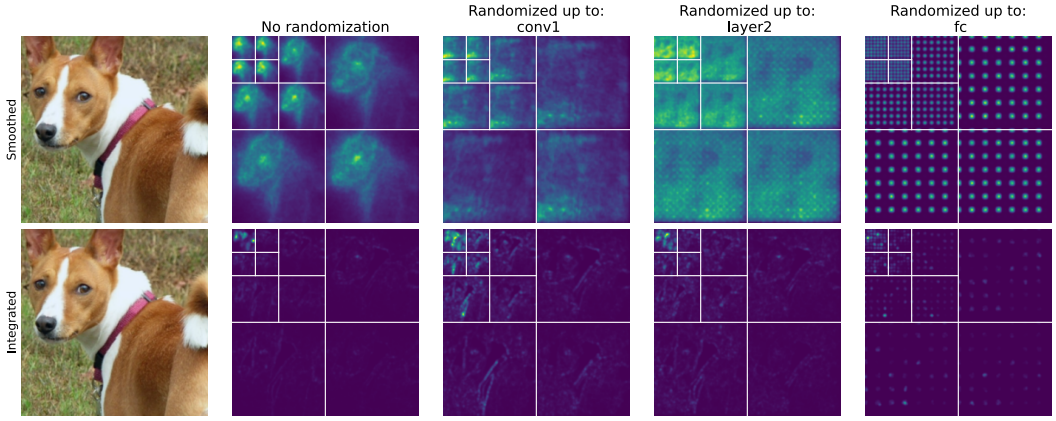

Figure 26: Illustration of the randomization test as a cascade of randomizations of the layers of the classifier. From left to right shows the explanation from the WAM for an increasingly randomized ResNet-18.

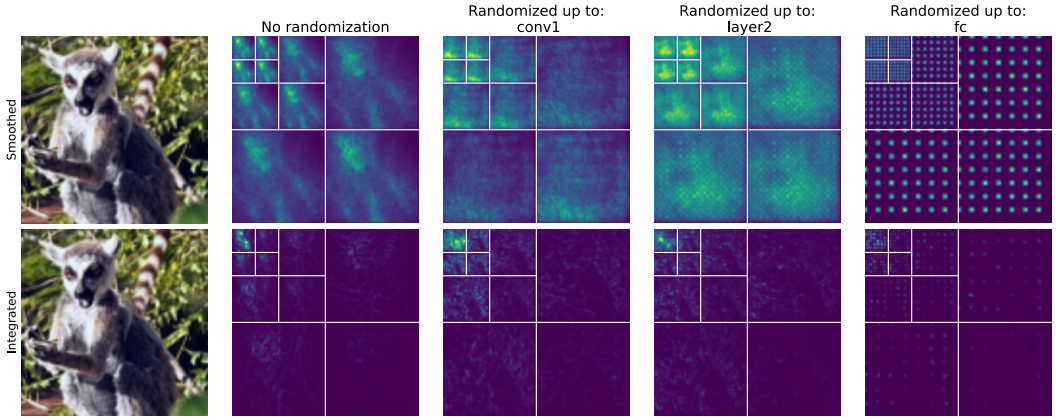

Figure 27: Illustration of the randomization test as a cascade of randomizations of the layers of the classifier. From left to right shows the explanation from the WAM for an increasingly randomized ResNet-18.

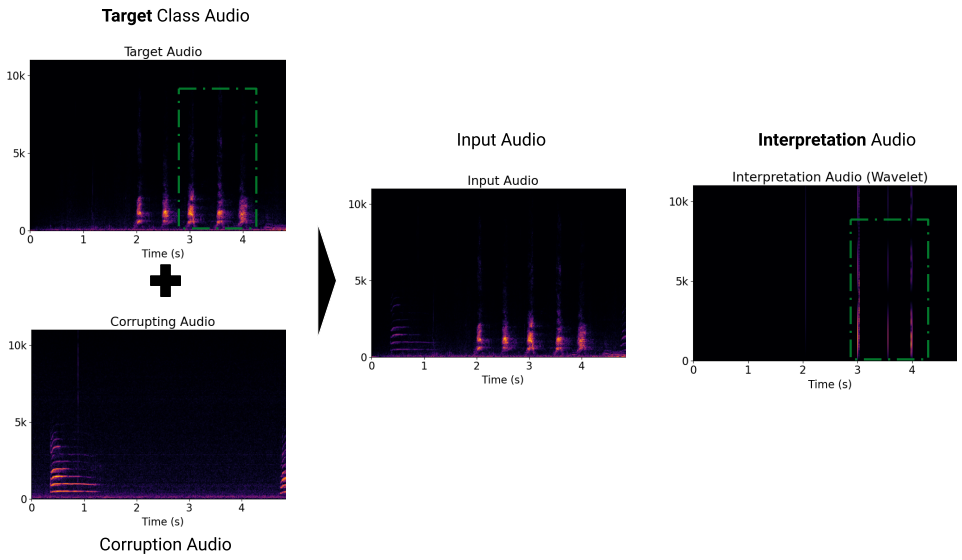

Figure 28: Qualitative illustration of WAM for audio via an Overlap experiment. The audio of the target class ('Dog') is mixed with a corrupting audio ('Cat') to form the input to the classifier. Interpretation audio reconstructed with important wavelet coefficients virtually eliminates signal from the corrupting audio and clearly emphasizes parts of the target class audio (indicated with green boxes).

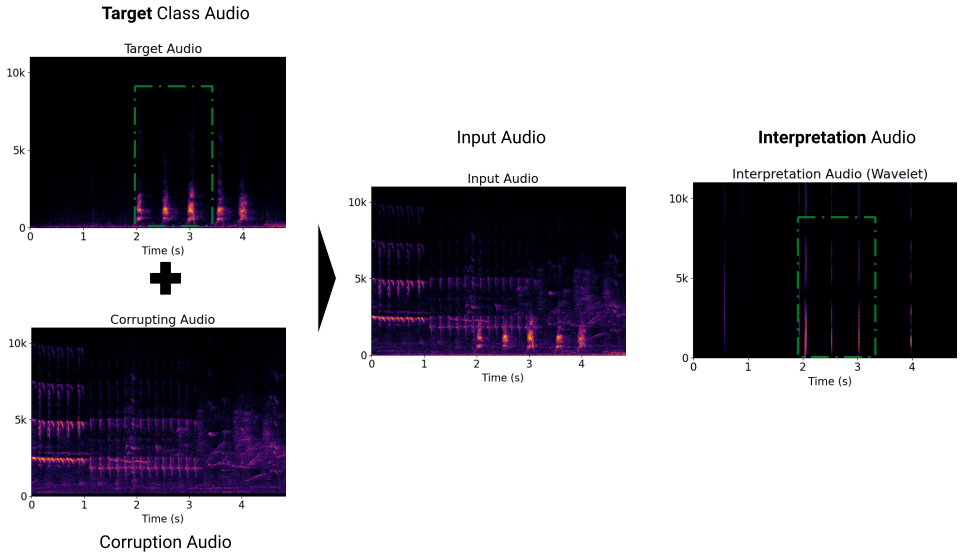

Figure 29: Qualitative illustration of WAM for audio via an Overlap experiment. The audio of the target class ('Dog') is mixed with a corrupting audio ('Rooster') to form the input to the classifier. Interpretation audio reconstructed with important wavelet coefficients virtually eliminates signal from the corrupting audio and clearly emphasizes parts of the target class audio (indicated with green boxes).