Listenable Maps for Audio Classifiers

Francesco Paissan [1 2] **Mirco Ravanelli** [3 2] **Cem Subakan** [4 3 2]

Abstract

Despite the impressive performance of deep learning models across diverse tasks, their complexity poses challenges for interpretation. This challenge is particularly evident for audio signals, where conveying interpretations becomes inherently difficult. To address this issue, we introduce Listenable Maps for Audio Classifiers (L-MAC), a posthoc interpretation method that generates faithful and listenable interpretations. L-MAC utilizes a decoder on top of a pretrained classifier to generate binary masks that highlight relevant portions of the input audio. We train the decoder with a loss function that maximizes the confidence of the classifier decision on the masked-in portion of the audio while minimizing the probability of model output for the masked-out portion. Quantitative evaluations on both in-domain and out-of-domain data demonstrate that L-MAC consistently produces more faithful interpretations than several gradient and masking-based methodologies. Furthermore, a user study confirms that, on average, users prefer the interpretations generated by the proposed technique.

1. Introduction

In recent years, deep learning models made significant strides in a variety of speech/audio applications, including sound event recognition, sound generation, speech recognition, speech separation, and many more (Ravanelli et al., 2021). An overwhelming majority of these models remain opaque concerning the interpretation of their predictions, as their large number of parameters, non-linearity, and high dimensionality make them "black-box" models (Molnar, 2022). Explainable Machine Learning is a research area that aims to render the models transparent concerning their decision-making mechanisms. Posthoc interpretability methods (Smilkov et al., 2017; Simonyan et al., 2014; Parekh et al., 2022), a sub-field within this domain, focus on generating interpretations for pre-trained machine learning models. These interpretations should ideally remain as faithful as possible to the pre-trained model while being easy to understand for humans. Many existing posthoc interpretability methods are primarily designed for computer vision, where the task often involves classifying objects against a clean background. In these cases, the interpretation commonly takes the form of a saliency map, which highlights the regions of the image relevant to model predictions. However, in the audio domain, achieving easily understandable interpretations poses a much greater challenge. State-of-the-art models for speech and audio processing typically operate on less interpretable inputs, such as mel-spectrograms, as compared to standard images. Consequently, generating saliency maps on these input features does not yield straightforward interpretations. A potentially more promising but relatively underexplored alternative involves generating listenable interpretations, which offer a more natural and user-friendly way for humans to comprehend the model prediction.

This paper contributes to this emerging field by introducing a novel method called Listenable Maps for Audio Classifiers (L-MAC). L-MAC outputs listenable explanations for pretrained audio classifiers that utilize mel-spectrograms or any other feature as input. Our approach employs a decoder that leverages information from the latent representations of pretrained classifiers to generate binary masks highlighting relevant audio segments. The decoder applies the mask not directly to the specific input features of the pretrained classifier but to the magnitude of Short-Time Fourier Transform (STFT) of the original input audio waveform. By inheriting the phase from the original signal, we can perform the Inverse Short-Time Fourier Transform (ISTFT), allowing the generation of a listenable waveform as the outcome of the interpretation process. The decoder is trained with a loss function that maximizes the confidence of the classifier's decision on the masked-in portion of the audio while minimizing the probability of the model output for the masked-out portion. Our loss term explicitly guides the interpreter to produce explanations that closely follow the target source without sacrificing the faithfulness of the explanations.

[1]Fondazione Bruno Kessler [2]Mila-Québec AI Institute [3]Concordia University [4]Laval University. Correspondence to: Francesco Paissan <fpaissan@fbk.eu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

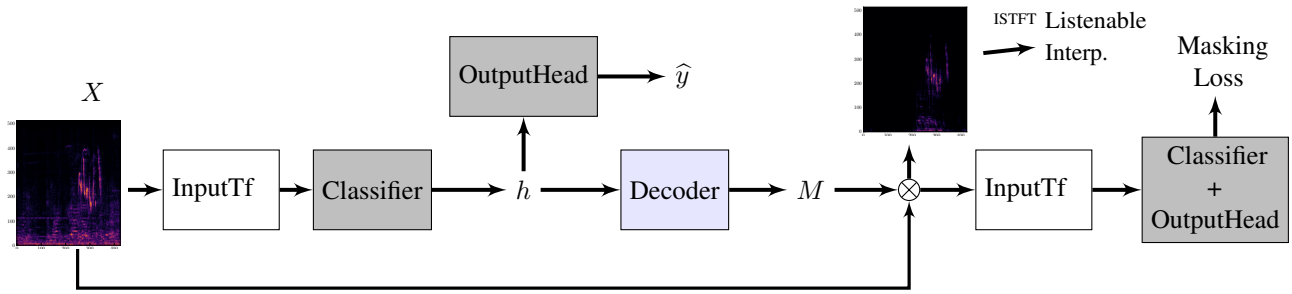

Figure 1. L-MAC Architecture. First, the linear spectrogram X is computed from an audio waveform x. Then, the audio features used by the pretrained classifier (e.g., FBANKs) are extracted (Input Tf). The classifier generates class predictions $\hat{y}$ and its latent representations h are input to the decoder, which produces a binary mask M for selecting specific portions of the original linear spectrogram X. The listenable interpretation is generated by applying the Inverse Short-Time Fourier Transform (ISTFT) on the masked spectrogram $X \odot M$ with the phase inherited from the original audio waveform. The masked loss used to train the decoder is computed based on the classifier predictions on the masked spectrogram and the predicted class $\hat{y}$.

To summarize, our contributions are the following:

- We propose a masking-based posthoc interpretation method for audio classifiers capable of providing listenable interpretations, even when the input audio is in the logmel domain.
- Using various faithfulness metrics, we demonstrate that our proposed method outperforms several existing posthoc interpretation methods on both in-domain and out-of-domain data.
- We conducted a user study, highlighting that users, on average, prefer the interpretation provided by L-MAC.
- We illustrate that our proposed method allows fine-tuning during training to improve audio quality. We show that this does not result in substantial loss in faithfulness.

1.1. Related Works

In the literature, various methods have been proposed for generating saliency maps, which aim to highlight the portions of the input that significantly contribute to the model prediction. These methods can be broadly categorized into masking-based and gradient-based approaches.

Gradient-based methods include techniques such as the standard saliency method, where a map is obtained by computing the gradient with respect to the input of the network, as described in (Simonyan et al., 2014). Other methods include integrated gradients (Sundararajan et al., 2017), guided back-prop (Springenberg et al., 2015), GradCAM (Selvaraju et al., 2019), and SmoothGrad (Smilkov et al., 2017). In (Adebayo et al., 2020), it has been shown that gradient-based approaches may not accurately capture the behavior of the classifier. Even under network weight and label randomization, these approaches essentially yield a behavior comparable to edge detection.

An emerging alternative is represented by masking-based approaches, which are the focus of this paper. Masking-based approaches involve estimating a mask (typically binary) to select a portion of the image that maximally contributes to the classifier decision. There exists some methods which directly learn a mask (Fong & Vedaldi, 2017; 2018; Petsiuk et al., 2018; Chang et al., 2019). In another line of work, the proposed methods involve training a decoder to estimate the mask, in the same vein as we do in this paper. These works include (Dabkowski & Gal, 2017; Fan, 2017; Zolna et al., 2020; Phang et al., 2020).

In the audio domain, notable works on interpretability include (Becker et al., 2023), which proposed layer-wise relevance propagation to generate saliency maps over spectrograms. Another noteworthy works include (Trinh et al., 2018; Kavaki & Mandel, 2020) where authors learn to identify important parts of the input spectrogram by masking additive white noise, within the context of automatic speech recognition. Additionally, (Won et al., 2019) proposed creating visualizations using attention layer outputs, while (Muckenhirn et al., 2019) suggested using Guided Backpropagation for spectrogram saliency maps. SLIME (Mishra et al., 2017; 2020) proposes to divide the spectrogram into predefined time/frequency regions (akin to the superpixels in LIME (Ribeiro et al., 2016) for images) and determines the feature importance for each region. AudioLIME (Haunschmid et al., 2020; Chowdhury et al., 2021), on the other hand, defines the LIME superpixels as sources extracted from the input audio and determines a saliency score for each source. More recently (Parekh et al., 2022) proposed Listen-to-Interpret (L2I) to learn the classifier relevance for Non-Negative Matrix factorization (NMF) (Lee & Seung, 1999) dictionaries, via a decoder trained to esti-

mate NMF activations. This work is particularly relevant to our method, as we also train a decoder. Consequently, we include detailed comparisons with L2I in our experiments.

2. Methodology

The architecture of the proposed L-MAC method is illustrated in Figure 1. Starting with the original audio waveform, we compute the (linear) spectrogram, denoted as X. The audio spectrogram X is then processed by a feature extraction block that computes the features needed by the pretrained classifier. Notably, the pretrained model can internally use various input features, such as FBANKs (mel-spectrogram features), making our approach feature-agnostic. These features are input to the classifier, which then generates predictions.

To generate an interpretation signal that explains these predictions, the latent representations h of the classifier are fed into the decoder, as illustrated in Figure 1. A more detailed diagram of how the classifier representations are used is shown in Figure 5 in Appendix A. The decoder is trained to generate a binary mask M, selecting relevant portions of the original spectrogram X. Importantly, the decoder is not specifically tuned to the features used by the pretrained classifier as we apply the binary mask to the (linear) spectrogram of the original audio X. This feature-agnostic aspect allows L-MAC to maintain listenability. By inheriting the phase from the original audio, we can indeed invert the masked spectrogram and produce a listenable audio signal as an interpretation outcome. To train the decoder, we feed the masked input into the classifier, and compute the mask loss. The objective is to learn a decoder capable of outputting a mask that accurately selects the region of interest in the input spectrogram. The main components of L-MAC are detailed in the following sub-sections.

2.1. The Masking Objective

The masking loss employed in this work draws inspiration from similar objectives described in (Dabkowski & Gal, 2017; Zolna et al., 2020; Phang et al., 2020). The goal is to maximize the confidence of the classification decision for the masked-in portion of the audio while minimizing it for the masked-out portion. The overall objective function is the following:

$$\min_{M} \alpha \mathcal{L}_{in}(f(M \odot X), y) - \beta \mathcal{L}_{out}(f((1-M) \odot X), y) + R(M), \quad (1)$$

where $f(.)$ represents the pretrained classifier being interpreted. The term $\mathcal{L}_{in}$ represents the categorical cross-entropy loss computed when we input the masked input $X \odot M$ to the pretrained classifier. In contrast to the aforementioned prior works, the categorical cross-entropy is computed using the network decision as the label, denoted as $y = \arg\max_c f_c(X)$, rather than the actual targets. Our objective is to minimize this term, as we want the masked signal to capture the elements that influenced the decision made by the classifier.

The term $\mathcal{L}_{out}$ represents the categorical cross-entropy computed for all parts of the input x not selected by the mask M. We aim to maximize it, as we want the mask to exclude information relevant to the pretrained classifier. Overall, we engage in a optimization problem where the $\mathcal{L}_{in}$ term tends to encourage larger masks, while the $\mathcal{L}_{out}$ term encourages smaller ones. The decoder must find a valuable trade-off between these aspects. Finally, note that $R(M_\theta(h))$ is a regularization term that includes an $l1$-regularizer to promote sparsity in the estimated mask.

One important aspect of our work is the use of a neural network, specifically the decoder, to estimate the binary mask M. This choice is motivated by our observation that neural networks yield more faithful and understandable masks. After parameterizing the interpretation mask M with a neural decoder, represented by parameters denoted as θ, the corresponding optimization objective becomes the following,

$$\min_{\theta} \lambda_{in} \mathcal{L}_{in}(\log f(M_\theta(h) \odot X), y) - \lambda_{out} \mathcal{L}_{out}(\log f((1 - M_\theta(h)) \odot X), y) + R(M_\theta(h)), \quad (2)$$

where the decoder M_θ maps an internal representation h of the classifier to a binary mask.

After the initial mask optimization, this framework allows a fine-tuning stage where the interpretation mask is refined to enhance the quality of the interpretations. This is accomplished by adding a term to the regularizer $R(.)$ as follows:

$$R(M_\theta(h)) = \lambda_g \|M_\theta(h) \odot X - X\| + \lambda_s \|M_\theta(h)\|_1,$$

where λ_g and λ_s are regularization coefficients, and X represents the spectrogram of the original signal. The first term in the regularization encourages the decoder to produce masked representations close to the original inputs, while the second term promotes sparsity in the mask. In our best configuration, this guidance is applied exclusively during training in a selective manner. Specifically, we only apply it to data items if the mask after the initial stage is highly similar to the binarized target spectrogram X. The similarity is measured by calculating the normalized cosine similarity between these two objects. This selective fine-tuning helps prevent steering the masks away from faithful interpretations. Finally, note that in case data augmentation is used, the target is chosen as the clean signal in the fune-tuning stage.

2.2. Producing Listenable Explanations

State-of-the-art audio and speech classifiers often rely on features computed on top of the linear spectrogram, such as FBANKs (mel-spectrograms). These features intelligently compress the frequency axis, often leading to more compact features that enhance performance. However, these features are non-invertible due to the compression applied to the frequency axis.

In our pipeline (Figure 1), we tackle this challenge by having the decoder $M_\theta(.)$ output a mask for the linear spectrogram X instead of generating a mask for the specific features used by the pretrained classifier. During training, we convert this masked signal $M_\theta(h) \odot X$ back to the feature domain before computing the training loss in Equation (2). The audio domain interpretations are obtained by inverting the linear spectrogram using the phase of the original audio waveform x, expressed as:

$$x_{\text{interpretation}} = \text{ISTFT}\left((M_\theta(h) \odot X)e^{jX_{\text{phase}}}\right). \quad (3)$$

3. Experiments

In our experiments, we evaluate the faithfulness and understandability of the generated interpretations. To achieve this, we considered two setups: i) Classification under in-domain conditions, and ii) Classification under out-of-domain conditions. We utilized the ESC50 dataset (Piczak) which contains 50 environmental sound classes for both setups. We also provide additional results on the UrbanSound8k dataset (Salamon et al., 2014) in Appendix B.

3.1. Metrics

To measure the faithfulness of classifications we have used the following metrics:

Faithfulness on Spectra (FF): This metric is originally introduced in (Parekh et al., 2022), as a way to measure how important is the generated interpretation for a classifier. The metric is calculated by measuring the drop in class-specific logit value, when the masked out portion of the interpretation mask is input to the classifier. This amounts to calculating,

$$\text{FF}_n := f(X_n)_c - f(X_n \odot (1 - M_\theta(h)))_c$$

If this metric is large, this signifies that the masked-in portion of the input spectrogram X is highly influential for the classifier decision for class c. We report the average faithfulness over all examples by reporting the average quantity $\text{FF} := \sum_n \frac{1}{N}\text{FF}_n$. Larger is better.

Average Increase (AI): Average increase, originally proposed in (Chattopadhay et al., 2018), measures the increase in confidence for the masked-in portion of the interpretation, and it is calculated as follows:

$$\text{AI} := \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}_{[f(X_n \odot M_\theta(h)) > f(X_n)_c]} \cdot 100,$$

where $\mathbf{1}_{[.]}$, is an indicator function which returns one if the argument is true, and zero otherwise. For this metric, larger is better.

Average Drop (AD): Average drop, originally proposed in (Chattopadhay et al., 2018), measures how much confidence is lost when the input image is masked, and calculated as follows:

$$\text{AD} := \frac{1}{N}\sum_{n=1}^{N} \frac{\max(0, f(X_n)_c - f(X_n \odot M_\theta(h))_c)}{f(X_n)_c} \cdot 100.$$

For this metric, smaller is better.

Average Gain (AG): This metric is first proposed in (Zhang et al., 2023), and it measures how much confidence is gained after masking the input image. It is calculated as follows:

$$\text{AG} := \frac{1}{N}\sum_{n=1}^{N} \frac{\max(0, f(X_n \odot M_\theta(h))_c - f(X_n)_c)}{1 - f(X_n)_c} \cdot 100.$$

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

$$\text{Fid-In} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}_{[\arg\max_c f(X_n)_c = \arg\max_c f_c(X_n \odot M_\theta(h))]}.$$

Larger values are better.

Sparseness (SPS): Sparseness measure is introduced in (Chalasani et al., 2020), and it measures if only values with large predicted saliency contribute to the prediction of the neural network. Larger values indicate more sparse/concise saliency maps. We use the implementation from the Quantus library (Hedström et al., 2023).

Complexity (COMP): Complexity metric is introduced in (Bhatt et al., 2020), and this metric measures the entropy of the distribution of contributions from each feature to the attibution. Smaller values indicate less complex interpretations. We again used the implementation from the Quantus library.

3.2. Faithfulness Evaluation

In these experiments, we first train a CNN14 classifier (Kong et al., 2020) on the ESC-50 dataset (Piczak) augmented with WHAM! noise, to simulate real-world mixtures. The classifier is trained on folds 1, 2, and 3 and obtains 75% and 78%

Table 1. In-domain quantitative evaluation on the ESC50 dataset. Our results reveal that L-MAC consistently achieves significantly higher faithfulness scores (AI, AD, AG, FF, Fid-In) compared to other methods.

	Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)
STFT- Mel	Saliency	0.00	15.79	0.00	0.05	0.07	0.39	5.48
	Smoothgrad	0.00	15.71	0.00	0.03	0.05	0.42	5.32
	IG	0.25	15.45	0.01	0.07	0.13	0.43	5.11
	GradCAM	8.50	10.11	1.47	0.17	0.33	0.34	5.64
	Guided GradCAM	0.00	15.61	0.00	0.05	0.06	0.44	5.12
	Guided Backprop	0.00	15.66	0.00	0.05	0.06	0.39	5.47
	L2I, RT=0.2	1.63	12.78	0.42	0.11	0.15	0.25	5.50
	L2I, RT=0.4	1.13	11.72	0.15	0.08	0.11	0.23	4.41
	L2I, RT=0.60	0.50	7.90	0.05	0.04	0.06	0.14	2.61
	L2I, RT=0.80	0.13	3.21	0.01	0.01	0.02	0.06	**0.98**
	SHAP	0.00	15.79	0.00	0.05	0.06	0.43	5.24
	L-MAC (ours)	**36.25**	**1.15**	**23.50**	0.20	**0.42**	**0.47**	4.71
	L-MAC, FT, $\lambda_g = 4$ (ours)	32.37	1.98	18.74	**0.21**	0.41	0.43	5.20
	L-MAC, FT, $\lambda_g = 16$ (ours)	27.12	3.32	16.18	0.19	0.39	0.44	5.03
	L-MAC, FT, $\lambda_g = 32$ (ours)	23.00	4.42	12.63	0.18	0.37	0.45	4.92
Mel	Saliency	0.00	15.81	0.00	0.10	0.07	0.39	4.53
	Smoothgrad	0.00	15.61	0.00	0.07	0.04	0.39	4.54
	IG	0.00	15.55	0.00	0.12	0.08	0.42	4.36
	GradCAM	7.00	10.93	1.04	0.17	0.29	0.34	**4.72**
	Guided GradCAM	0.125	15.40	6.67	0.08	0.07	**0.45**	4.17
	Guided Backprop	0.125	15.54	0.00	0.10	0.08	0.39	4.53
	SHAP	0.00	15.57	0.00	0.11	0.08	0.41	4.42
	L-MAC (ours)	35.63	1.59	**24.28**	0.22	**0.42**	**0.45**	4.11
	L-MAC (ours) FT, $\lambda_g = 4$	**36.13**	**1.28**	21.15	**0.23**	**0.42**	0.32	4.71

classification accuracy on folds 5 and 4, respectively. The CNN14 classifier we employed has 12 2D convolutional layers and is pre-trained on the VGG-sound dataset (Chen et al., 2020a) using SimCLR (Chen et al., 2020b). The decoder of L-MAC consists of a series of transposed 2D convolutional layers. Each convolutional layer upsamples the time and frequency axis. The classifier's representations are fed to the decoder in a U-Net-like fashion to incorporate information at different time-frequency resolutions (as shown in Figure 5 in Appendix A). Specifically, the decoder takes the four deepest representations of the classifier.

We then freeze the weights of the classifier and train a decoder on the same training set as the classifier (ESC50 + WHAM! noise), as shown in Figure 1. We perform two sets of evaluations on the ESC-50 dataset to validate the robustness of L-MAC in real-world settings. First, we evaluate L-MAC on in-domain data, where the interpreter is tested against data similar to the one in the training set (ESC 50 folds 4 and 5 with WHAM! noise). The results are provided in Table 1. In the second place, we evaluate L-MAC performance on out-of-domain data. We generate out-of-domain data by creating mixtures of samples from folds 4 and 5. The results are provided in Table 2. We report results obtained with STFT and Mel domain spectra for in-domain and out-of-domain data. We have also reported additional results on the UrbanSound8k dataset (Salamon et al., 2014) in Table 2, in Appendix B.

To compare L-MAC with the literature, we used several gradient-based methods such as standard saliency maps (Simonyan et al., 2014), SmoothGrad (Smilkov et al., 2017), IntegratedGradients (Sundararajan et al., 2017), GuidedBackProp (Springenberg et al., 2015), and decoder based audio specific method, Listen-to-Interpret (L2I) (Parekh et al., 2022), and we also include SHAP (Lundberg & Lee, 2017). For L2I, we have reported results using relevance thresholds RT=0.2, 0.4, 0.6, and 0.8 (an important hyperparameter for the L2I method). We have used the Captum implementations (Kokhlikyan et al., 2020) for the gradient-based methods and SHAP and adapted the SpeechBrain (Ravanelli et al., 2021) implementation for L2I. For L-MAC, we have obtained results for finetuning strengths $\lambda_g = 4, 16, 32$. The implementation of our experimental setup is accessible through the companion website.[1].

[1] https://fpaissan.github.io/lmac

Table 2. Out-of-Domain Quantitative Evaluation for the ESC50 Dataset. In out-of-distribution conditions, L-MAC consistently outperforms other methods across all evaluated metrics.

	Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)
STFT- Mel	Saliency	0.62	31.73	0.07	0.06	0.12	0.76	11.06
	Smoothgrad	0.12	31.84	0.00	0.06	0.13	0.83	10.66
	IG	0.37	31.15	0.03	0.12	0.26	0.87	10.22
	L2I	5.00	25.65	1.00	0.20	0.35	0.52	10.99
	GradCAM	14.12	17.62	7.46	0.25	0.00	0.91	9.66
	Guided GradCAM	0.00	31.74	0.00	0.07	0.11	0.89	10.24
	Guided Backprop	0.63	31.73	0.07	0.06	0.11	0.76	11.06
	SHAP	0.00	31.81	0.00	0.07	0.14	0.84	10.58
	L-MAC (ours)	**60.63**	**4.82**	**35.85**	**0.39**	**0.81**	**0.94**	**9.61**
	L-MAC FT, $\lambda_g = 4$ (ours)	50.75	6.73	26.00	**0.39**	0.78	0.84	10.51
	L-MAC FT, $\lambda_g = 16$ (ours)	37.62	10.67	19.29	0.34	0.70	0.87	10.19
	L-MAC - FT, $\lambda_g = 32$ (ours)	28.88	12.69	14.56	0.32	0.66	0.89	10.01
	L-MAC - FT, $\lambda_g = 4$ (ours), CCT 0.7	52.87	6.71	29.46	0.38	0.78	0.93	9.76
	L-MAC - FT, $\lambda_g = 16$ (ours), CCT 0.7	45.87	8.12	23.91	0.37	0.74	0.91	9.93
	L-MAC - FT, $\lambda_g = 32$ (ours), CCT 0.7	38.50	9.62	19.11	0.35	0.70	0.89	10.04
Mel	Saliency	0.38	31.64	0.01	0.15	0.12	0.77	9.17
	Smoothgrad	0.25	31.66	0.01	0.14	0.11	0.79	9.03
	IG	0.12	31.52	0.01	0.19	0.19	0.84	8.62
	GradCAM	19.88	18.85	4.67	0.34	0.69	0.66	9.49
	Guided GradCAM	0.00	31.68	0	0.14	0.12	0.89	10.24
	Guided Backprop	0.38	31.64	0.01	0.15	0.12	0.77	9.16
	SHAP	0.25	31.60	0.00	0.17	0.15	0.82	8.81
	L-MAC (ours)	60.25	**4.84**	**34.72**	**0.44**	0.80	**0.90**	**8.29**
	L-MAC - FT, $\lambda_g = 4$ (ours)	**60.75**	**4.84**	29.34	**0.44**	**0.83**	0.64	9.38
	L-MAC - FT, $\lambda_g = 16$ (ours)	45.75	9.93	17.04	0.43	0.80	0.69	9.16
	L-MAC - FT, $\lambda_g = 32$ (ours)	37.50	8.65	14.08	0.43	0.77	0.70	9.05

For in-domain data, we observe in Table 1 that L-MAC generally results in better faithfulness scores (AI, AD, AG, FF, Fid-In) compared to the baselines. Since we work with a classifier trained in the Mel domain, we would like to note that we evaluated the methods in cases where we both mask in the STFT domain (denoted with STFT-Mel in Tables 1, 2) and in the Mel domain (denoted with Mel in Tables 1, 2). This is important, as masking in the STFT domain gives the ability to produce listenable interpretations. We observe that for L-MAC, the STFT domain interpretations result in similar faithfulness scores compared to the Mel domain. More specifically, for the in-domain evaluation, we observe that the gradient-based methods such as standard saliency, Smoothgrad, IG, GradCAM, Guided BackProp, and SHAP generally result in less faithful interpretations than the decoder-based methods such as L2I and L-MAC. We see that the most faithful results are obtained without having the finetuning (FT) stage in L-MAC. However, with additional finetuning, we see that L-MAC can produce results that only lose slightly from their faithfulness scores while increasing the understandability of the interpretation, as shown by user preference. We observe that after finetuning, L-MAC can still be more faithful compared to the other baselines we investigate, including our L2I implementation. We also see that in terms of the sparsity metric, L-MAC has better overall numbers than L2I and the other gradient-based methods. In terms of the complexity metric, it is comparable with the gradient-based methods and with L2I with a relevance threshold of 0.2. Note that with larger relevance threshold values, L2I returns less active interpretations.

In Table 2, we show the metrics on out-of-distribution data. In this case, the gradient-based posthoc saliency methods also do not yield very faithful results except for GradCAM. We also observe that L-MAC outperforms L2I even if a severe finetuning strength of $\lambda_g = 32$ is employed. Another observation is that the listenable STFT version of L-MAC yields comparable faithfulness results to the direct Mel domain interpretations yielded by L-MAC. For this data, we also try increasing the cross-correlation threshold (denoted with CCT in Table 2) between the interpretations and the target mask during training (as discussed in Section 2). We see that using a larger CCT generally helps increase the interpretations' faithfulness for larger λ_g values.

3.3. Qualitative Evaluation

In order to assess the quality of the evaluations, we have conducted a user study with 15 participants to evaluate the perceived quality of the produced interpretations. We gave the evaluators the following two instructions:

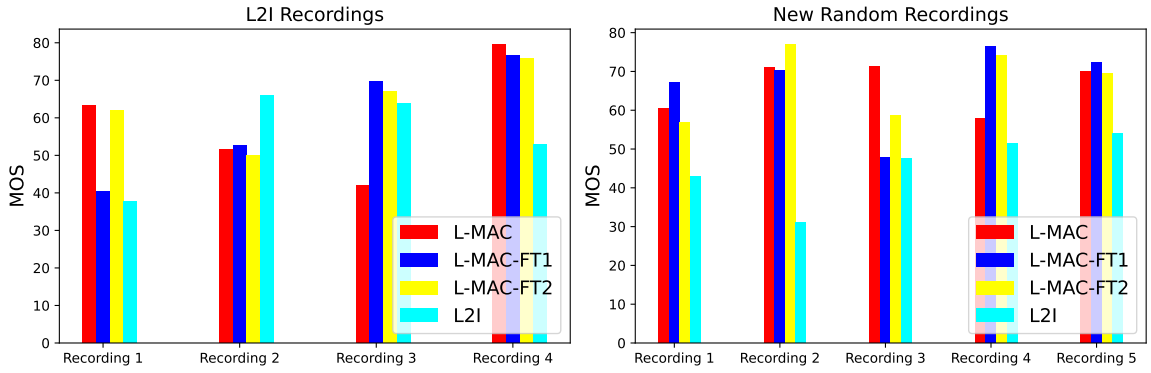

Figure 2. The Mean Opinion Scores (MOS) obtained in the user study. **(Left)** MOS values obtained on recordings from L2I companion website **(Right)** MOS values obtained on newly created random recordings with two sound classes.

1. How well does the interpretation correspond to the part of the input audio associated with the given class?
2. While evaluating, please pay attention to audio quality also.

Note that we showed the label of the predicted class to the participants. We have asked the users to rate the interpretations they listen between 0-100. We have used the open source webMushra (Schoeffler & et al., 2018) package.

In order to directly compare the perceived quality of the interpretations of L-MAC and L2I we used the audio samples provided in the L2I companion website. That is, we have downloaded the first four audio tracks and the corresponding generated interpretations from the official companion website[2] of L2I. These audio tracks are similar to the audio tracks we have used for OOD evaluation in Section 3.2, as they are formed by mixing two audio recordings. We show the summary of this user study in Table 3 in the MOS-1 column. The results indicate that, on average, users preferred the quality of the interpretations provided by L-MAC compared to the interpretations provided in the companion website of L2I. We also show the comparison of mean-opinion-scores specific to each recording in the left panel of Figure 2. We observe that for each recording, L-MAC interpretations result in either better or comparable preference. It is worth noting that, in general, fine-tuning improves user preference compared to the standard L-MAC.

In addition to the showcase recordings for L2I, we have also randomly picked five mixture recordings that we have created and presented these recordings to the users. In total therefore we have presented each user 9 recordings. The average preference for these recordings are shown in Table 3 in the MOS-2 column. Once again, we observe that L-MAC surpasses L2I, and fine-tuning further enhances user preference. The results of this user study for each recording

[2] https://jayneelparekh.github.io/listen2interpret/

Table 3. Mean-Opinion Scores for the showcased examples from the L2I Website. L-MAC is, on average, preferred by the users.

Method	MOS-1	MOS-2
L-MAC	59.13	64.00
L-MAC, FT, $\lambda_g = 16$	59.8	66.7
L-MAC, FT, $\lambda_g = 16$, CCT0.7	**63.7**	**67.2**
L2I	55.1	50.8

are depicted in the right panel of Figure 2. Some examples of interpretations can be found at our companion website[3].

3.4. Sanity Checks

Beyond the quantitative and qualitative evaluation of L-MAC, we have also conducted two sanity checking experiments. First, we tested L-MAC against the Remove-and-retrain (ROAR) test proposed in (Hooker et al., 2019). This test checks the classification accuracy when top-k percent of the most time-frequency bins which are deemed to be the most salient are removed, and then a classifier is retrained on the remaining portion of the input spectra. Intuitively, this sanity check verifies whether the interpreter focuses on relevant time-frequency points for the class of interest. We see in the left panel of the Figure 4 that with increasing percentage in the removed portion, compared to randomly removing the masks the classification accuracy drops faster with L-MAC interpretations, which suggests that L-MAC returns interpretations on semantically relevant portion of the spectra.

We have also conducted the cascading randomization test from (Adebayo et al., 2020) in order to verify that L-MAC does not generate interpretations that are invariant to the classifier weights, but actually is sensitive to the randomization of the classifier weights. In Figure 3, we showcase this, where we compare the interpretations obtained when the

[3] https://fpaissan.github.io/lmac/

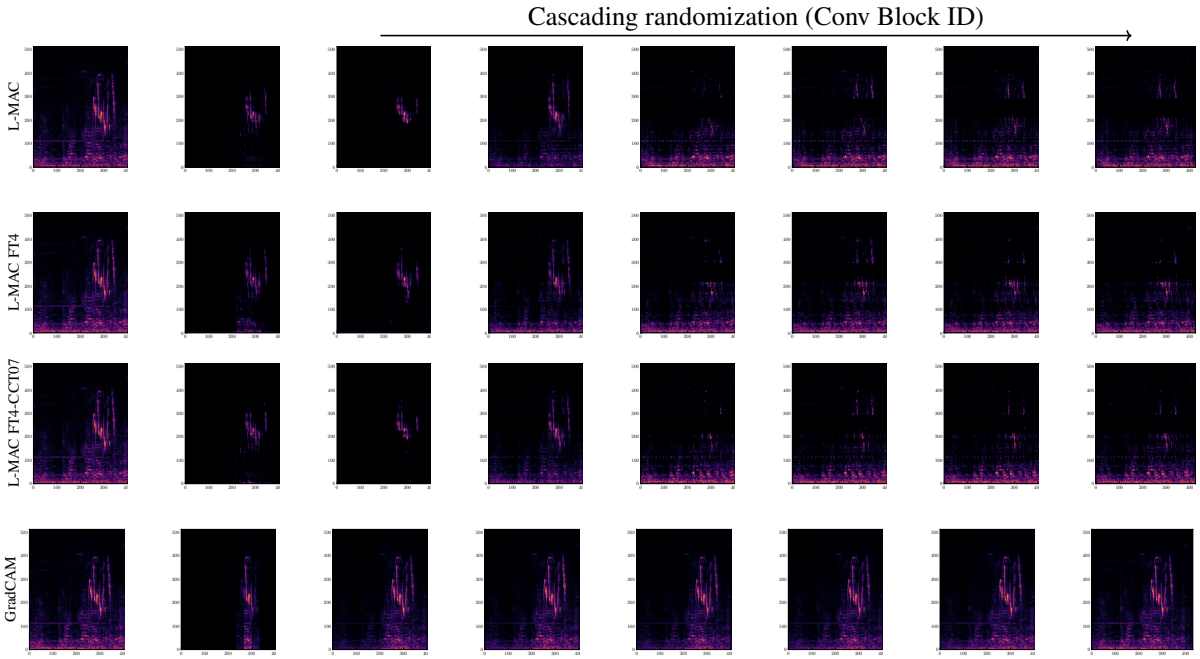

Figure 3. Example demonstrating the behaviour of L-MAC during the MRT test. From left to right: original sample, interpretation, and interpretations generated by randomizing the weights of the convolutional blocks starting from the logits in a cascading fashion, as suggested in (Adebayo et al., 2020). As expected, the interpretations are corrupted by randomizing the weights of the model. From top to bottom: L-MAC, L-MAC finetuned with $\lambda_g = 4$ and CCT$= 0.6$, L-MAC finetuned with $\lambda_g = 4$ and CCT $= 0.7$, and GradCAM.

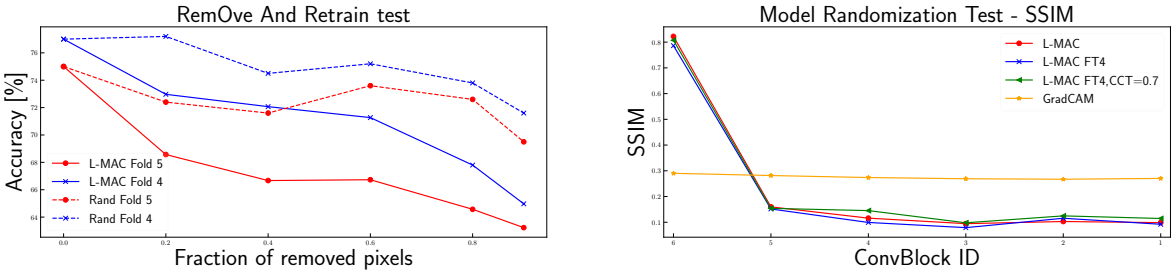

Figure 4. Sanity checks for saliency maps: (left) RemOve And Retrain test. The presented results are the averages over three runs. The dashed line represents the random attribution baseline. (right) Structured Similarity Index (SSIM) extracted using the Model Randomization Test.

classifier layers are randomized. We see that, as expected, with more random layers L-MAC interpretations lose their original focus. Whereas for instance with GradCAM interpretations, we observe that the interpretations basically remain insensitive to the classifier weight randomization. We furthermore quantify this finding by calculating the Structural Similarity Index (SSIM) between the interpretations generated for the original model and the interpretations generated for the model with randomized weights (as was done in (Adebayo et al., 2020)). Similar to the visualization of the interpretations in Figure 3, we see that L-MAC interpretations quicly drop in similarity after starting from the 5'th convolutional block (5'th deepest block - note that we start randomizing from the, last layer, and then go down). Whereas, we see that the GradCAM interpretations remain practically unchanged.

3.5. Additional Results on Out-of-Domain Data

In addition to the Out-of-Domain experiments conducted in Section 3.2, we have also tested L-MAC on audio samples corrupted with white noise and speech. We have created 3dB Signal-to-noise ratio mixtures, and used samples from the LJSpeech (Ito & Johnson, 2017) dataset for speech. In Table 4, we show these additional results on ESC50 dataset. In Appendix B with Table 5, we provide the results for this experimental setup applied on the UrbanSound8K benchmark. We observe that L-MAC is able to obtain better results in terms of quantitative faithfulness metrics such as AI, AD, AG, FF, and Fid-In. In this table we also report the mask-mean to indicate the size of the obtained mask (denoted with MM). We observe that even though the mask area of L-MAC is significantly smaller than GradCAM for instance, it is able to obtain considerably better metrics that

Table 4. Additional results obtained on the ESC50 dataset with White Noise and LJSpeech contamination. The masking is done on the STFT domain for this experiment. Note that we also indicate the mask-mean (denoted with MM)

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	Classification on ESC50, White Noise Contamination, 38.6% accuracy							
Saliency	0.25	26.31	0.02	0.05	0.06	0.79	10.92	0.016
Smoothgrad	0.00	26.37	0.00	0.04	0.09	0.84	10.62	0.01
IG	0.75	25.60	0.56	0.10	0.21	0.82	10.65	0.01
L2I @ 0.2	0.00	19.41	0.21	0.11	0.04	**36.62**	**7.32**	0.12
GradCAM	8.87	20.88	1.24	0.28	0.51	0.69	11.25	0.18
Guided GradCAM	0.50	26.23	0.05	0.07	0.11	0.91	10.14	0.01
Guided Backprop	0.25	26.30	0.02	0.05	0.07	0.79	10.92	0.02
SHAP	0.12	26.34	0.001	0.05	0.12	0.86	10.40	0.004
L-MAC (ours)	**83.62**	**1.50**	**56.12**	**0.33**	**0.86**	0.92	10.03	0.06
All-ones baseline	0	0	0	0.34	1	N.A.	N.A.	1
	Classification on ESC50, LJSpeech Contamination, 79.3% accuracy							
Saliency	0.87	26.00	0.20	0.06	0.11	0.75	11.10	0.02
Smoothgrad	0.50	26.14	0.11	0.05	0.13	0.79	10.91	0.08
IG	0.37	25.70	0.01	0.11	0.25	0.87	10.14	0.00
L2I @ 0.2	1.75	29.49	0.27	0.15	0.18	0.79	**9.56**	0.16
GradCAM	20.37	13.49	2.63	0.28	0.73	0.66	11.33	0.22
Guided GradCAM	0.25	26.10	0.09	0.06	0.11	0.88	10.30	0.01
Guided Backprop	0.87	26.01	0.20	0.05	0.11	0.75	11.10	0.02
SHAP	0.00	26.14	0.00	0.06	0.16	0.79	10.81	0.01
L-MAC (ours)	**70.75**	**2.73**	**39.64**	**0.33**	**0.83**	**0.93**	9.70	0.05
All-ones baseline	0	0	0	0.35	1	N/A	N/A	1

measure faithfulness. Moreover, we have also calculated the scores obtained with an all-zeros mask. We observe that the FF score obtained with L-MAC is very similar to the all-ones mask which removes the entire spectrogram during the score computation, and we also observe that the mask-mean of L-MAC is significantly smaller than 1. This indicates that the masked-out portion of the L-MAC interpretations are not very significant for the classifier.

4. Conclusions

This paper introduced a novel approach, called Listenable Maps for Audio Classifiers (L-MAC), that produces post-hoc interpretations for audio classifiers. L-MAC employs a decoder that uses the latent representations of the black-box classifier to estimate a binary mask that effectively highlights the portions of the input audio that triggered the prediction made by the classifier. The application of this mask to the linear-frequency-scale spectrogram enables generating interpretations that are listenable. We would like to note that our methodology is also applicable for classifiers whose input domains are not linear-frequency-scale spectra. We train the decoder using an objective that promotes faithfulness to the classifier decisions, that minimizes the categorical cross-entropy for masked inputs while maximizing it for masked-out inputs.

Through an extensive experimental evaluation involving quantitative and qualitative assessments, as well as through sanity checks, our results demonstrate that L-MAC achieves significantly superior faithfulness metrics and user preference, compared to various baselines.

Impact statement

This paper proposes a methodology to improve neural network interpretability. We do not foresee a direct, socially negative impact. On the contrary, the proposed methodology may facilitate socially beneficial applications of machine learning-based audio processing by enhancing the human trust in neural network decisions.

Acknowledgements

We thank Geraldin Nanfack and Eugene Belilovsky for insightful discussions. This research was enabled in part by support provided by Calcul Québec and the Digital Research Alliance of Canada.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps, 2020.

Becker, S., Vielhaben, J., Ackermann, M., Müller, K.-R., Lapuschkin, S., and Samek, W. Audiomnist: Exploring explainable artificial intelligence for audio analysis on a simple benchmark, 2023.

Bhatt, U., Weller, A., and Moura, J. M. F. Evaluating and aggregating feature-based model explanations, 2020.

Chalasani, P., Chen, J., Chowdhury, A. R., Jha, S., and Wu, X. Concise explanations of neural networks using adversarial training, 2020.

Chang, C.-H., Creager, E., Goldenberg, A., and Duvenaud, D. Explaining image classifiers by counterfactual generation, 2019.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, March 2018.

Chen, H., Xie, W., Vedaldi, A., and Zisserman, A. Vgg-sound: A large-scale audio-visual dataset. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020b.

Chowdhury, S., Praher, V., and Widmer, G. Tracing back music emotion predictions to sound sources and intuitive perceptual qualities, 2021.

Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers, 2017.

Fan, L. Adversarial localization network. 2017. URL https://api.semanticscholar.org/CorpusID:52087996.

Fong, R. and Vedaldi, A. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks, 2018.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. doi: 10.1109/iccv.2017.371. URL http://dx.doi.org/10.1109/ICCV.2017.371.

Haunschmid, V., Manilow, E., and Widmer, G. audiolime: Listenable explanations using source separation, 2020.

Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., and Höhne, M. M. M. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. URL http://jmlr.org/papers/v24/22-0142.html.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks, 2019.

Ito, K. and Johnson, L. The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

Kavaki, H. S. and Mandel, M. I. Identifying Important Time-Frequency Locations in Continuous Speech Utterances. In *Proc. Interspeech 2020*, pp. 1639–1643, 2020. doi: 10.21437/Interspeech.2020-2637.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch, 2020.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 2020.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. URL https://api.semanticscholar.org/CorpusID:4428232.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions, 2017.

Mishra, S., Sturm, B. L., and Dixon, S. Local interpretable model-agnostic explanations for music content analysis. In *International Society for Music Information Retrieval Conference*, 2017. URL https://api.semanticscholar.org/CorpusID:795766.

Mishra, S., Benetos, E., Sturm, B. L., and Dixon, S. Reliable local explanations for machine listening, 2020.

Molnar, C. *Interpretable Machine Learning*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

Muckenhirn, H., Abrol, V., Magimai-Doss, M., and Marcel, S. Understanding and Visualizing Raw Waveform-Based CNNs. In *Proc. Interspeech 2019*, pp. 2345–2349, 2019. doi: 10.21437/Interspeech.2019-2341.

Paissan, F., Subakan, C., and Ravanelli, M. Posthoc interpretation via quantization, 2023.

Parekh, J., Parekh, S., Mozharovskyi, P., d'Alché-Buc, F., and Richard, G. Listen to interpret: Post-hoc interpretability for audio networks with nmf. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35270–35283, 2022.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models, 2018.

Phang, J., Park, J., and Geras, K. J. Investigating and simplifying masking-based saliency methods for model interpretability, 2020.

Piczak, K. J. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. Speechbrain: A general-purpose speech toolkit, 2021.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier, 2016.

Salamon, J., Jacoby, C., and Bello, J. P. A dataset and taxonomy for urban sound research. In *ACM International Conference on Multimedia*, 2014.

Schoeffler, M. and et al. webmushra — a comprehensive framework for web-based listening tests. 2018.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, (2):336–359, October 2019.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise, 2017.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net, 2015.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks, 2017.

Trinh, V. A., McFee, B., and Mandel, M. I. Bubble Cooperative Networks for Identifying Important Speech Cues. In *Proc. Interspeech 2018*, pp. 1616–1620, 2018. doi: 10.21437/Interspeech.2018-2377.

Won, M., Chun, S., and Serra, X. Toward interpretable music tagging with self-attention. 2019. URL http://arxiv.org/abs/1906.04972.

Zhang, H., Torres, F., Sicre, R., Avrithis, Y., and Ayache, S. Opti-cam: Optimizing saliency maps for interpretability, 2023.

Zolna, K., Geras, K. J., and Cho, K. Classifier-agnostic saliency map extraction. *Computer Vision and Image Understanding*, 196:102969, July 2020. ISSN 1077-3142. doi: 10.1016/j.cviu.2020.102969. URL http://dx.doi.org/10.1016/j.cviu.2020.102969.

A. Decoder structure illustration

In Figure 5 we show the architecture of the decoder $M_\theta(\cdot)$ employed in our experiments.

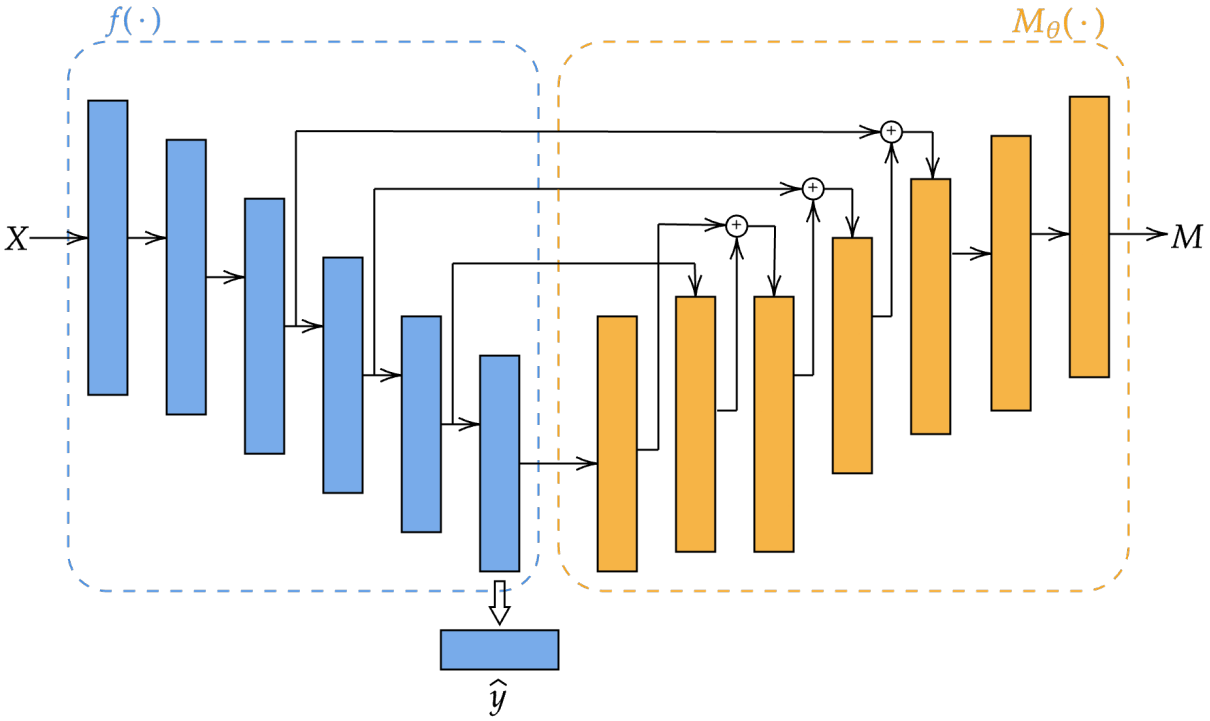

Figure 5. Diagram of the decoder neural network $M_\theta(\cdot)$. The representations from the classifier $f(\cdot)$ are fed through different layers of the decoder $M_\theta(\cdot)$ via skip connections.

B. Results on UrbanSound8k Dataset

We provide additional results for the In-Domain and Out-of-Domain experiments on the UrbanSound8k dataset (Salamon et al., 2014), in addition to the experiments conducted on the ESC50 dataset in the mainbody of the paper. In Table 5, we first report the results obtained on the In-Domain experiments, and then for the Out-of-Domain experiments where we contaminate the samples with other samples from the US8k dataset, White Noise, and speech (the same experimental setup as the setup on the ESC50). We observe that on both the ID experiment and on different OOD experiments, L-MAC is able to outperform or obtain very similar results compared to the baselines. We also observe that L-MAC masks are in general smaller compared to the ones obtained by comparably performing models such as GradCAM or L2I, which suggests that L-MAC is able to obtain masks that are not trivial.

Table 5. Additional experiments on UrbanSound8k Dataset

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	Classification on US8k, WHAM! Noise Contamination (ID experiment), 82.8% accuracy							
Saliency	0.40	23.86	0.26	0.072	0.12	0.30	5.69	0.02
Smoothgrad	0.00	26.86	0.00	0.04	0.04	0.30	5.69	0.01
IG	0.25	23.70	0.15	0.10	0.12	**0.40**	**5.24**	0.005
L2I @ 0.2	0.55	19.59	0.28	0.17	0.18	0.24	5.67	0.21
GradCAM	7.68	6.94	4.04	0.31	0.39	0.13	5.96	0.29
Guided GradCAM	0.40	24.29	0.26	0.06	0.10	0.33	5.58	0.01
Guided Backprop	0.40	23.86	0.26	0.07	0.12	0.30	5.69	0.02
SHAP	0.10	26.32	0.03	0.05	0.06	0.33	5.59	0.01
L-MAC (ours)	**19.09**	**2.51**	**13.57**	**0.40**	**0.46**	0.32	5.55	0.14
All-ones baseline	0	0	0	0.42	1	N/A	N/A	1
	Classification on US8k, US8k Contamination, 82.6% accuracy							
Saliency	0.40	48.24	0.23	0.10	0.24	0.61	11.35	0.056
Smoothgrad	0.00	54.65	0.00	0.06	0.09	0.60	11.40	0.023
IG	0.51	46.31	0.20	0.15	0.30	**0.80**	**10.39**	0.01
L2I @ 20	2.88	27.84	2.01	0.27	0.38	0.50	11.05	0.37
GradCAM	14.65	9.71	6.42	0.61	0.86	0.18	12.02	0.60
Guided GradCAM	0.40	49.61	0.19	0.10	0.21	0.66	11.17	0.03
Guided Backprop	0.40	48.24	0.23	0.10	0.24	0.61	11.35	0.04
SHAP	0.05	53.66	0.01	0.07	0.12	0.67	11.17	0.01
L-MAC (ours)	**27.12**	**9.58**	**20.04**	**0.77**	**0.85**	0.71	10.75	0.24
All-ones baseline	0	0	0	0.86	1	N/A	N/A	1
	Classification on US8k, White Noise Contamination, 48.5% accuracy							
Saliency	0.56	48.42	0.16	0.13	0.14	0.57	11.50	0.04
Smoothgrad	0.00	48.96	0.00	0.07	0.10	0.66	11.24	0.02
IG	0.85	47.92	0.30	0.15	0.16	0.63	11.21	0.02
L2I @ 0.2	27.45	32.73	23.45	0.25	0.42	0.32	11.81	0.41
GradCAM	**37.37**	16.69	**31.29**	0.45	0.69	0.24	11.95	0.61
Guided GradCAM	0.51	48.26	0.27	0.12	0.14	0.62	11.31	0.03
Guided Backprop	0.56	48.41	0.16	0.13	0.13	0.59	11.44	0.04
SHAP	0.15	48.51	0.02	0.09	0.12	**0.72**	**10.98**	0.01
L-MAC (ours)	31.62	**16.31**	22.4	**0.81**	**0.71**	0.39	11.72	0.39
All-ones baseline	0	0	0	0.74	1	N/A	N/A	0
	Classification on US8k, LJSpeech Contamination, 88.7% accuracy							
Saliency	0.30	50.60	0.11	0.08	0.25	0.58	11.44	0.04
Smoothgrad	0.00	52.68	0.00	0.05	0.19	0.61	11.39	0.02
IG	0.35	50.13	0.18	0.10	0.26	**0.79**	**10.43**	0.01
L2I @ 0.20	1.21	41.91	0.72	0.23	0.27	0.55	10.50	0.23
GradCAM	13.29	**7.16**	6.20	0.53	**0.91**	0.25	11.93	0.61
Guided GradCAM	0.30	51.18	0.14	0.08	0.23	0.64	11.27	0.03
Guided Backprop	0.30	50.6	0.11	0.09	0.25	0.58	11.44	0.04
SHAP	0.05	52.15	0.01	0.06	0.18	0.67	11.16	0.01
L-MAC (ours)	**18.18**	9.91	**11.28**	**0.90**	0.86	0.69	10.91	0.26
All-ones baseline	0	0	0	0.83	1	N/A	N/A	0