Listenable Maps for Zero-Shot Audio Classifiers

Francesco Paissan[*1,2,5], **Luca Della Libera**[2,3], **Mirco Ravanelli**[2,3], **Cem Subakan**[2,3,4]
[1]Fondazione Bruno Kessler, [2]Mila, Québec AI Institute, [3]Concordia University,
[4]Laval University, [5] University of Trento

Abstract

Interpreting the decisions of deep learning models, including audio classifiers, is crucial for ensuring the transparency and trustworthiness of this technology. In this paper, we introduce LMAC-ZS (Listenable Maps for Zero-Shot Audio Classifiers), which, to the best of our knowledge, is the first decoder-based post-hoc explanation method for explaining the decisions of zero-shot audio classifiers. The proposed method utilizes a novel loss function that aims to closely reproduce the original similarity patterns between text-and-audio pairs in the generated explanations. We provide an extensive evaluation using the Contrastive Language-Audio Pretraining (CLAP) model to showcase that our interpreter remains faithful to the decisions in a zero-shot classification context. Moreover, we qualitatively show that our method produces meaningful explanations that correlate well with different text prompts.

1 Introduction

The widespread adoption of AI in critical decision-making processes makes interpreting the decisions of deep learning models crucial for ensuring transparency and trustworthiness. Recently, significant research has been devoted to explainable machine learning [1]. These efforts aim to either employ interpretable models or explain the decisions of black-box models using posthoc explanation methods. In the audio domain, however, only a few works exist on interpretable audio classifiers [2, 3, 4] as well as on posthoc explanation methods [5, 6, 7, 8]. The latter contributions are limited to standard closed-set classification and do not explore the challenging topic of interpreting zero-shot classifiers. Zero-shot classifiers, on the other hand, are gaining popularity for their exceptional adaptability, as they define audio classes based on a set of textual prompts [9]. The class labels are not necessarily predefined but can be generated dynamically during inference via natural language. The increased flexibility of zero-shot classifiers comes with a drawback: their predictions are challenging to interpret. This difficulty arises from their multi-modal nature, as learning an interpreter in the joint representation space between text and audio is required. A notable example of a zero-shot classifier is Contrastive Language Audio Pretraining (CLAP) [10], which jointly trains audio and text representations using contrastive learning, that we also work with in this paper.

This paper addresses the problem of posthoc explanations for zero-shot audio classifiers. To the best of our knowledge, this has never been attempted before in the literature. Following the masking idea proposed in [8], we propose LMAC-ZS (Listenable Maps for Audio Classifiers in the Zero-Shot context), which consists of a decoder (the interpreter) that outputs a saliency map capable of highlighting the regions within the input audio that trigger the zero-shot classification. We introduce a novel loss function that incentives faithfully following the similarity between the original audio and the corresponding text prompt. Our method provides listenable explanations for linear and non-linear frequency-scale short-time Fourier transform (STFT) representations of audio waveforms. It can also operate on the raw audio domain directly. We applied our explanation method on top of a pretrained version of the popular CLAP [10] by considering different zero-shot classification datasets, including

[*]Correspondance to fpaissan@fbk.eu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2087

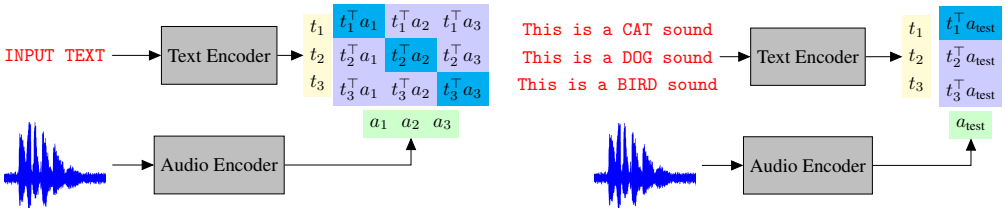

Figure 1: **(left)** The training of the CLAP model for learning cross-modal representations. **(right)** Zero-shot classification with the CLAP model.

the ESC50 [11], UrbanSound8K [12], as well as versions of ESC50 and UrbanSound8K where different types of contaminations are applied. We show extensive experimental results suggesting that the produced saliency maps correlate well with the corresponding text prompts and faithfully follow the original zero-shot classifier. In particular, our evaluation using various faithfulness metrics highlights that LMAC-ZS is able to provide explanations that are highly relevant to the decisions made by the CLAP model in the zero-shot context. Our method significantly outperforms traditional approaches such as GradCAM++ [13], highlighting their inefficiency in challenging tasks such as zero-shot audio classification.

In summary, our contributions are the following:

- We propose a new method, LMAC-ZS, to explain zero-shot audio classifiers.
- We show that LMAC-ZS maintains faithfulness to the CLAP predictions across diverse zero-shot scenarios.
- We qualitatively show that LMAC-ZS produces meaningful explanations for different text prompts.

1.1 Related Work

Posthoc explanation methods aim to explain the decisions of pretrained neural networks. Several works exist on producing posthoc explanations with gradient-based approaches in the computer vision literature. These include the standard saliency method [14], GradCAM [15], GradCAM++ [13], SmoothGrad [16], Integrated Gradients (IG) [17], and several others. However, as suggested in [18], these methods often fail to follow the classifier very faithfully and tend to be insensitive even to random model weights. Another category of post-hoc explanation methods in computer vision generates explanations by applying masks to the input data. Key approaches in this category include [19, 20, 21, 22], which use optimization-based techniques to learn and generate these masks. There also exists a series of works that are most closely related to this paper, where a decoder is trained to produce explanations. Notable attempts in this vein include Dabkowski and Gal (2017) [23], Fan et al. (2017) [24], Zolna et al. (2020) [25], and Phang et al. (2020).

In the audio domain, several post-hoc explanation methods exist. These methods employ various techniques such as layer-wise relevance propagation [26], guided backpropagation [27], and LIME [28, 29, 6, 30]. More recent posthoc explanation methods that use a decoder to produce masks on spectrograms include Listen-to-Interpret [5], which uses a Non-Negative Matrix Factorization [31] based decoder to produce non-negative saliency maps. Other examples include Posthoc Interpretation via Quantization [7], which trains a VQ-VAE [32]-based decoder as an explanation module, and Listenable Maps for Audio Classifiers [8], which trains a decoder using a classification loss to promote faithfulness. These works are not directly applicable to zero-shot classification as they require a predefined set of labels to train the interpreter. In this paper, our goal is to produce explanations in a true zero-shot fashion. To achieve that, we train our decoder on the same data as the CLAP model (without using class labels that we will later test on). Subsequently, LMAC-ZS can produce explanations for arbitrary labels, encoded as natural language. This includes labels not previously seen during the training of the interpreter.

2 Preliminaries

In this Section, we first present the learning methodology for audio-text cross-modal representations in Section 2.1. Then, we introduce masking-based posthoc explanations in Section 2.2.

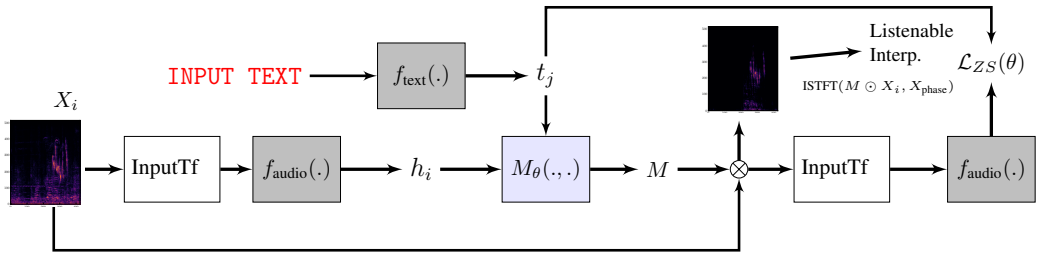

Figure 2: LMAC-ZS architecture. The input spectrogram (linear frequency) X_i (the i-th audio in the batch) first of all passes through the transformations (InputTf block) to make it compatible with the input domain (e.g. Mel Spectra) of the audio encoder $f_{\text{audio}}(.)$, which yields the latent representations h_i. These representations along with the text representation t_j (the j-th text prompt within the batch) are then fed to the decoder $M_\theta(.,.)$. The resulting mask is then element-wise multiplied with the input spectrogram X_i. The masked spectrogram $M \odot X_i$ is then converted back to the input domain of the audio encoder, and the similarity score $t_i^\top f_{\text{audio}}\big(M_\theta(t_i, h_j) \odot X_{\text{audio},j}\big)$ is calculated, which is used in the overall training objective $\mathcal{L}_{ZS}(\theta)$. The listenable explanation is produced by simply inverting the masked spectrogram through the inverse-STFT by incorporating the phase spectrogram of the input X_{phase}.

2.1 Contrastive Learning of Audio-Text Cross-Modal Representations

The goal of learning audio-text cross-modal representations is to create a joint latent space between text and audio. CLAP (Contrastive Language-Audio Pretraining) [10], achieves this via contrastive learning. That is, the similarity between the latent representations of a text and audio signal is maximized if they form a pair, otherwise this similarity is minimized. More specifically, consider X_t and X_a as batches of text and audio data, respectively. Within the CLAP model, the latent representation is derived by passing the text and audio through their respective encoders, denoted as $g_t(.)$ and $g_a(.)$. This process produces the text and audio latent representations, denoted as $L_{\text{text}} = g_{\text{text}}(X_{\text{text}})$ and $L_{\text{audio}} = g_{\text{audio}}(X_{\text{audio}})$, respectively. Here, L_{text} is a matrix of dimensions $\mathbb{R}^{N \times T}$, where N is the batch size and T represents the latent dimensionality of text. Similarly, L_{audio} is a matrix of dimensions $\mathbb{R}^{N \times A}$, where A denotes the latent dimensionality of audio. CLAP trains a joint latent space by passing L_{text} and L_{audio} through fully-connected layers such that,

$$t = \textbf{MLP}_{\text{text}}(L_{\text{text}}),\ a = \textbf{MLP}_{\text{audio}}(L_{\text{audio}}), \tag{1}$$

where $\textbf{MLP}(.)$ denotes the multi-layer perceptron transformation layers. The matrix $t \in \mathbb{R}^{N \times d}$ and $a \in \mathbb{R}^{N \times d}$ respectively denote the text and audio latent variables with the same latent dimensionality d. As a shorthand for the rest of the paper we will denote the combination of encoders and the MLP with $f_{\text{text}}(.) := \textbf{MLP}_{\text{text}}(g_{\text{text}}(.))$ and $f_{\text{audio}}(.) := \textbf{MLP}_{\text{audio}}(g_{\text{audio}}(.))$ for text and audio, respectively. The model aims to maximize the diagonal entries on the matrix $C = ta^\top$. The matrix $C \in \mathbb{R}^{N \times N}$ represents audio-text pairings. The diagonal elements $C_{i,i}$ correspond to positive samples, while other elements are negative samples. This translates into the following training loss function:

$$\mathcal{L}(C) = -\frac{1}{2}\sum_{i=1}^{N}\Big(\log \textbf{Softmax}_t(C/\tau)_{i,i} + \log \textbf{Softmax}_a(C/\tau)_{i,i}\Big), \tag{2}$$

where $\textbf{Softmax}_t(.)$ and $\textbf{Softmax}_a(.)$ respectively denote Softmax functions along text and audio dimensions, τ is a temperature scaling parameter, and the $C_{i,i}$ denotes the diagonal elements of the C matrix. We show the training forward pass pipeline in the left panel of Figure 1.

We would like to note that with this framework, the zero-shot classification amounts to calculating the similarity of the representation of a given audio with a set of text prompts, each corresponding to a class labels. Namely, the classification decision is taken as:

$$\widehat{c} = \arg\max_j t_j^\top a_{\text{test}} = \arg\max_j f_{\text{text}}(\texttt{prompt}_j)^\top f_{\text{audio}}(X_{\text{audio}}^{\text{test}}), \tag{3}$$

where $\widehat{c}$ is the zero-shot classification decision, a_{test} is the embedding for the test audio, and t_j is the text embedding corresponding to the label of class j (represented via $\texttt{prompt}_j$). We show the pipeline of zero-shot classification in the right panel of Figure 1.

https://doi.org/10.52202/079017-2087

2.2 Saliency Maps For Fixed Set Audio Classifiers

In this work, we adopt a posthoc explanation method that uses a learnable decoder, following the masking idea introduced in L-MAC [8]. Before we delve into how to generate a saliency map for a zero-shot classifier, we first explain how L-MAC produces a saliency map within the context of a standard classification setup. The loss function that is minimized during training in L-MAC to obtain faithful saliency maps is defined as follows:

$$\mathcal{L}(\theta) = \text{CrossEnt}(\widehat{y}; f\left(M_\theta(h) \odot X\right)) - \text{CrossEnt}(\widehat{y}, f\left((1 - M_\theta(h)) \odot X\right)) + \lambda \|M_\theta(h)\|_1. \quad (4)$$

The first term in this loss function aims to maximally align the classifier prediction $\widehat{y} = \arg\max_c f_c(X)$, with the classifier output obtained after masking the input, i.e. the logit $f\left(M_\theta(h) \odot X\right) \in \mathbb{R}^{N_C}$, where N_C is the number of classes. Note that CrossEnt(.) denotes the CrossEntropy loss function. The decoder network $M_\theta(h)$ takes in the classifier representations h (which can consist of representations of several layers) and produces a mask (with values within the interval $[0, 1]$ and same size as the input) that is element-wise multiplied with the input X. A regularization term that consists of an L_1 loss is also used to prevent trivial solutions, such as a mask with all values set to 1. The mask-out term $-\text{CrossEnt}(\widehat{y}, f\left((1 - M_\theta(h)) \odot X\right))$ minimizes the relevance of the mask-out portion to the predicted class $\widehat{y}$. In the next section, we introduce our framework for explaining zero-shot classifiers (that we have defined in Section 2.1), which again applies a mask to the input to replicate the original text-audio similarities.

3 Saliency Maps for Zero-Shot Audio Classifiers

Similarly to the method introduced in L-MAC [8] and summarized in Section 2.2, our goal is to generate explanations that faithfully follow the model. However, in the context of zero-shot classifiers, we do not have a model that outputs a fixed number of logits. Hence, we need a different loss function that promotes faithfulness between the explanations and the zero-shot audio classifier, which relies on similarities to make its decisions. We denote the similarity between the i-th text prompt and j-th audio recording with $C_{i,j}$ as,

$$C_{i,j} = t_i^\top a_j = t_i^\top f_{\text{audio}}(X_{\text{audio},j}). \quad (5)$$

Our methodology is based on obtaining a saliency map such that the text-audio cross-modal similarity matrix C is maximally preserved after masking the important parts of the spectrogram. In other words, we learn a decoder such that, after masking the audio, the similarity with text prompts within the batch is maximally preserved. To this end, we define the loss function as follows:

$$\begin{aligned}\mathcal{L}_{\text{ZS}}(\theta) = &\sum_{i,j} \left| C_{i,j} - t_i^\top f_{\text{audio}}\Big(M_\theta(t_i, h_j) \odot X_{\text{audio},j}\Big) \right| + \lambda_1 \Big\| M_\theta(t_i, h_j) \Big\|_1 \\ &+ \lambda_2 \sum_i D(X_{\text{audio},i}).\end{aligned} \quad (6)$$

Here, the first term aims to minimize the discrepancy between the original similarities $C_{i,j}$ and the similarities after masking the input audio $X_{\text{audio},j} \in \mathbb{R}^{T \times F}$ using the decoder $M_\theta(t_i, h_j)$, which outputs a mask of shape $T \times F$. Importantly, the decoder is conditioned on both the text representation $t_i = f_{\text{text}}(X_{\text{text},i})$ that corresponds to the i-th text prompt in the batch, and the representations h_j, which includes the last 4 representations obtained from the audio encoder $f_{\text{audio}}(X_{\text{audio},j})$. λ_1, λ_2 are tradeoff parameters.

The second term in Equation 6 promotes sparsity in the generated mask to avoid trivial solutions. Finally, the last term $D(.)$ aims to increase the diversity of masks generated for a given audio when conditioned on different text prompts. It is defined as:

$$D(X_{\text{audio},i}) = \sum_{j; j \neq i} \left\| t_i^\top t_j - f_{\text{audio}}\Big(X_{\text{audio},i} \odot M_\theta(t_i, h_i)\Big)^\top f_{\text{audio}}\Big(X_{\text{audio},i} \odot M_\theta(t_j, h_i)\Big) \right\|. \quad (7)$$

The goal of this term is to align the uni-modal similarity between text embeddings t_i, t_j with the uni-modal similarity between the corresponding audio embeddings $f_{\text{audio}}\big(X_{\text{audio},i} \odot M_\theta(t_i, h_i)\big)$, $f_{\text{audio}}\big(X_{\text{audio},i} \odot M_\theta(t_j, h_i)\big)$, obtained from the corresponding masked spectrograms. The intuition

is that the similarity between two text prompts should be reflected in the similarity of the audio embeddings from the corresponding masked spectrograms: the farther the text prompts, the farther apart should be the corresponding audio embeddings from masked spectrograms, and thus, the more different the masks should be. We show the effectiveness of this term on diversity with respect to different text prompts in Section B. The overall pipeline is shown in Figure 2.

Producing Listenable Explanations: Our method employs its masking in the linear Short-Time Fourier Transform (STFT) domain, and therefore generating listenable explanations through the inverse-STFT is possible. The listenable explanation is obtained through the following operation,

$$x_{\text{int}} = \text{ISTFT}\left((X \odot M)e^{jX_{\text{phase}}}\right), \tag{8}$$

where both the explanation mask M and the input audio X are in the linear-scale STFT domain, and X_{phase} is the phase of the original input audio. This operation is also shown in Figure 2.

4 Experiments

4.1 Metrics

To evaluate our method, we employ faithfulness metrics previously used in the audio interpretability literature for standard classification setups. We adapt such metrics to the zero-shot scenario by using the class prediction probabilities defined by audio-text similarities such that

$$p(c = j) = \frac{\exp(t_j^\top a_{\text{test}})}{\sum_{k=1}^{N_c} \exp(t_k^\top a_{\text{test}})}, \tag{9}$$

where $p(c = j)$ is the probability of predicting the class that corresponds to the j-th text prompt and N_c is the total number of text prompts used in the zero-shot setting. Analogously to CLAP [10], to create prompts that correspond to the predefined classes in ESC50 [11] and UrbanSound8K [12], we augment the class labels with the prefix *"this is the sound of"*, obtaining prompts such as *"this is the sound of baby crying"*, *"this is the sound of cat"*. When computing all the metrics for LMAC-ZS, we conditioned the decoder on the text prompt that corresponds to the model prediction $\widehat{c} = \arg\max t_j^\top a_{\text{test}}$.

Faithfulness on Spectra (FF): Introduced in [5], it assesses the importance of the provided explanation for the classifier. The metric is calculated by measuring how much does a class-specific prediction probability drops after removing the explanation signal from the original. It is defined as

$$\text{FF}_n = p_{\widehat{c}}(X_n) - p_{\widehat{c}}(X_n - X_{int}),$$

where $\widehat{c}$ is the class prediction given by the classifier. High faithfulness values mean that the masked-in portion of the input spectrogram X is highly influential for the classifier decision of the predicted class $\widehat{c}$. 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): Introduced in [13], it measures the increase in confidence for the masked-in portion of the explanation, and it is calculated as follows:

$$\text{AI} = \frac{1}{N}\sum_{n=1}^{N} [p_{\widehat{c}}(X_n \odot M) > p_{\widehat{c}}(X_n)] \cdot 100,$$

where $[.]$, is the indicator function, which is one if the argument is true, and zero otherwise. For this metric, larger is better.

Average Drop (AD): Introduced in [13], it measures the decrease in model confidence when the input image is masked, and it is calculated as follows:

$$\text{AD} = \frac{1}{N}\sum_{n=1}^{N} \frac{\max(0, p_{\widehat{c}}(X_n) - p_{\widehat{c}}(X_n \odot M))}{p_{\widehat{c}}(X_n)} \cdot 100.$$

For this metric, smaller is better.

 https://doi.org/10.52202/079017-2087

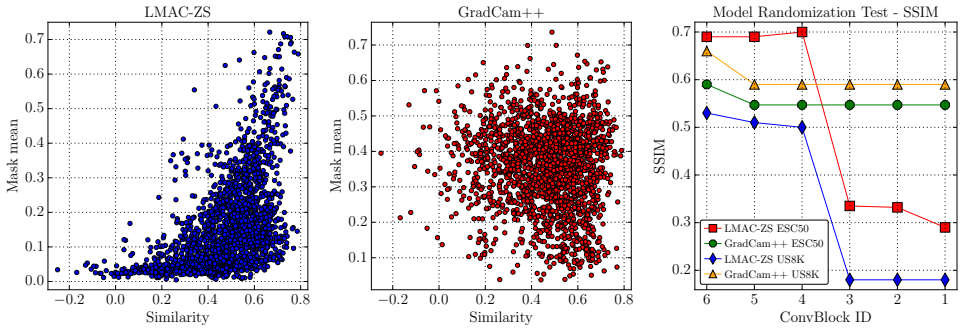

Figure 3: **(left)** Mask-Mean vs Similarity for LMAC-ZS, **(middle)** Mask-Mean vs Similarity for GradCam++, **(right)** Model Randomization Test for LMAC-ZS and GradCam++.

Average Gain (AG): Introduced in [33], it measures the increase in confidence after masking the input image. It is calculated as follows (larger is better):

$$\text{AG} = \frac{1}{N}\sum_{n=1}^{N}\frac{\max(0, p_{\hat{c}}(X_n \odot M) - p_{\hat{c}}(X_n))}{1 - p_{\hat{c}}(X_n)} \cdot 100.$$

Input Fidelity (Fid-In): Introduced in [7], it measures whether the classifier outputs the same class prediction on the masked-in portion of the input image. It is defined as the following and the larger is better,

$$\text{Fid-In} = \frac{1}{N}\sum_{n=1}^{N}[\arg\max_{c} p_c(X_n) = \arg\max_{c'} p_{c'}(X_n \odot M)].$$

Sparseness (SPS): Introduced in [34], it measures whether 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 [35].

Complexity (COMP): Introduced in [36], it measures the entropy of the distribution of contributions from each feature to the attribution. Smaller values indicate less complex explanations. We again use the implementation from the Quantus library.

4.2 Experimental Setup

We use the official pretrained CLAP [10] weights[2] to perform zero-shot classification on ESC50 [11] and UrbanSound8K [12] datasets. We train LMAC-ZS on the datasets on which CLAP had been trained (namely, Clotho [37], FSD50K [38], AudioCaps [39], and MACS [40] which are publicly available). We also explored training LMAC-ZS only on Clotho to simulate the case where the computational budget is limited. The models were trained on a single NVIDIA RTX 3090 GPU. For the LMAC-ZS model that is trained on the Clotho dataset, we did 2 epochs on the complete dataset, for which an epoch approximately takes an hour. For the Full CLAP data we did 2 epochs as well, and an epoch takes around 4 hours. We quantitatively test whether LMAC-ZS follows the zero-shot classifier on In-Domain (ID) and Out-of-Domain (OOD) settings. For the In-Domain setting, we perform zero-shot classification on clean audio from ESC50 and UrbanSound8k and then produce explanations for the classifications using LMAC-ZS. We would like to emphasize that LMAC-ZS has only been trained on the training datasets for CLAP, and has not been fine-tuned on ESC50 or UrbanSound. For the Out-of-Domain setting, we contaminate the audio with various noise sources at 3dB Signal-to-Noise Ratio (other audio from the same dataset, white-noise, and human speech from the LJ-Speech [41] dataset).

We explore masking in the Mel-domain to explore the case where we produce explanations directly in the feature space on which CLAP operates. For Mel-domain we used 44.1kHz data on which the CLAP model is trained. We also explore masking in the linear frequency-scale log power-STFT

[2] `https://zenodo.org/records/8378278`

Table 1: In-Domain quantitative evaluation for the ESC50 and UrbanSound8K Datasets. Two versions of LMAC-ZS are compared: (CT) trained on the Clotho dataset only and (Full) trained on all CLAP datasets. MM denotes the Mask-Mean, the average value for the obtained masks.

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	ZS classification on ESC50, Mel-Masking, 80.7% accuracy							
Gradcam	2.90	45.85	1.01	0.28	0.19	0.71	9.52	0.15
GradCam++	8.45	35.07	3.19	0.50	0.39	0.41	10.32	0.35
SmoothGrad	0.50	52.76	0.12	0.024	0.036	0.301	10.52	0.039
IG	0.25	53.47	0.054	0.064	0.022	0.57	10.09	0.037
LMAC-ZS (CT)	**29.00**	**12.25**	**12.93**	0.49	**0.80**	0.78	9.40	0.14
LMAC-ZS (Full)	23.45	17.12	10.31	**0.51**	0.68	**0.80**	**9.12**	0.17
	ZS classification on ESC50, STFT-Masking, 78.9% accuracy							
GradCam	20.30	23.75	7.77	0.78	0.58	0.72	11.54	0.14
GradCam++	32.50	8.97	7.95	0.79	0.84	0.41	12.41	0.35
SmoothGrad	6.95	32.75	2.85	0.78	0.47	0.53	11.98	0.0001
IG	16.10	21.51	6.05	**0.79**	0.65	**0.74**	11.58	0.0095
LMAC-ZS (CT)	37.40	7.43	**11.26**	0.78	0.86	0.50	**12.29**	0.11
LMAC-ZS (Full)	**43.35**	**4.29**	10.57	0.78	**0.90**	0.65	11.86	0.1
	ZS classification on US8K, Mel-Masking, 71.7% accuracy							
GradCam	2.34	47.55	1.09	0.26	0.16	0.78	9.32	0.12
GradCam++	7.21	33.4	3.33	**0.56**	0.44	0.41	10.27	0.39
SmoothGrad	1.21	49.68	0.43	0.04	0.11	0.33	10.49	0.04
IG	0.98	50.77	0.35	0.15	0.09	0.60	10.02	0.03
LMAC-ZS (CT)	23.41	20.58	12.88	0.51	0.65	**0.85**	9.01	0.08
LMAC-ZS (Full)	**35.69**	**15.65**	**18.19**	0.48	**0.72**	0.79	**8.95**	0.17
	ZS classification on US8K, STFT-Masking, 68.9% accuracy							
GradCam	18.67	26.1	11.18	0.79	0.53	0.77	11.41	0.12
GradCam++	32.85	8.84	13.16	0.81	0.83	0.41	12.34	0.39
SmoothGrad	15.31	23.56	7.67	**0.81**	0.61	0.54	11.97	0.0001
IG	22.65	19.53	12.31	0.77	0.66	**0.79**	11.36	0.01
LMAC-ZS (CT)	32.71	14.57	14.69	0.75	0.72	0.55	12.12	0.08
LMAC-ZS (Full)	**40.85**	**7.79**	**15.52**	0.78	**0.85**	0.76	**11.34**	0.07

domain to be able to provide listenable explanations. For STFT domain filtering we worked with 16kHz data. We would like to note that this results in slight changes in zero-shot classification accuracies, which are reported in the Tables 1, 2, 3. We trained LMAC-ZS with a batch size of 2 using the Adam optimizer [42] with a learning rate of 1e−5. The decoder consists of a series of transposed convolutions to upsample from CNN14 [43] CLAP representations and incorporates text conditioning by using cross-attention similar to that used in Stable Diffusion [44]. The implementation is done using the SpeechBrain toolkit [45, 46] and it can be accessed through [3].

4.3 Quantitative Comparison

We compare LMAC-ZS with popular gradient-based saliency map methods including GradCam [15], GradCam++ [13], SmoothGrad [16], and Integrated Gradients (IG) [17]. We apply these saliency map methods using only the CNN14 audio representations. The class logit with respect to which the class activation map for these methods is calculated is picked by using the zero-shot classification decision $\widehat{c} = \arg\max_j t_j^\top a_{\text{test}}$.

In Table 1, we compare the faithfulness of the explanations obtained on In-Domain data, where we performed zero-shot classification on clean ESC50 and US8k recordings. We observe that on ESC50 with Mel-Domain masking, LMAC-ZS obtains better AI, AD, AG, FF, and Fid-In values. We observe a similar trend for AI, AD, and AG with STFT-domain masking also, while FF values are comparable. On the UrbanSound8K dataset, we also observe that in terms of AI, AD, and AG the best results are obtained with LMAC-ZS trained with the Full CLAP training datasets. In terms of mask sparseness (SPS) and Complexity (COMP) in most cases, the best results are obtained with the proposed model.

In Table 2, we compare the faithfulness of the explanations obtained on ESC50 samples contaminated with three different types of background noises. We observe that with Mel-Masking, LMAC-ZS

[3] `https://francescopaissan.it/lmaczs`

https://doi.org/10.52202/079017-2087

Table 2: Out-of-Domain quantitative evaluation for the ESC50 Dataset.

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	ZS classification on ESC50, Mel-Masking, ESC50 contamination, 57.2% accuracy							
GradCam	6.78	40.71	3.13	0.29	0.19	0.69	9.66	0.18
GradCam++	9.82	35.81	4.53	**0.42**	0.29	0.39	10.40	0.35
SmoothGrad	0.62	48.55	0.13	0.024	0.022	0.29	10.54	0.039
IG	0.55	48.88	0.091	0.073	0.020	0.56	10.13	0.039
LMAC-ZS (CT)	19.25	24.30	8.83	0.40	0.49	0.81	9.18	0.13
LMAC-ZS (Full)	**20.43**	**21.57**	**9.71**	**0.42**	**0.54**	**0.82**	**9.08**	0.15
	ZS classification on ESC50, STFT-Masking, ESC50 contamination, 58.6% accuracy							
GradCam	23.77	25.25	12.24	0.69	0.49	0.69	**11.73**	0.17
GradCam++	29.52	14.84	10.17	**0.70**	0.70	0.39	12.48	0.35
SmoothGrad	11.80	30.63	5.15	**0.70**	0.42	0.52	12.06	0.0002
IG	16.37	25.67	7.21	**0.70**	0.51	**0.71**	11.73	0.011
LMAC-ZS (CT)	35.65	12.23	**13.04**	0.69	0.74	0.53	12.18	0.09
LMAC-ZS (Full)	**39.4**	**8.28**	11.81	0.69	**0.80**	0.67	11.79	0.09
	ZS classification on ESC50, Mel-Masking, White Noise contamination, 65.2% accuracy							
GradCam	3.65	43.79	1.43	0.34	0.12	0.75	9.41	0.14
GradCam++	7.12	37.03	2.97	**0.52**	0.26	0.43	10.33	0.335
SmoothGrad	1.72	47.93	0.56	0.040	0.040	0.28	10.54	0.035
IG	1.57	47.97	0.55	0.084	0.039	0.54	10.16	0.034
LMAC-ZS (CT)	**28.52**	**17.72**	**12.78**	0.42	**0.64**	0.82	9.18	0.19
LMAC-ZS (Full)	14.25	27.92	6.62	0.41	0.42	**0.86**	**8.86**	0.11
	ZS classification on ESC50, STFT-Masking, White Noise contamination, 57.4% accuracy							
GradCam	14.92	31.89	5.95	**0.66**	0.32	0.77	11.40	0.12
GradCam++	19.50	24.01	8.04	**0.66**	0.50	0.42	12.42	0.33
SmoothGrad	7.10	36.53	2.66	**0.66**	0.25	0.52	12.15	0.0004
IG	10.17	34.35	4.89	**0.66**	0.30	**0.69**	**11.80**	0.011
LMAC-ZS (CT)	19.85	21.51	7.13	0.63	0.53	0.52	12.24	0.08
LMAC-ZS (Full)	**32.97**	**11.86**	**10.63**	0.64	**0.70**	0.65	11.85	0.09
	ZS classification on ESC50, Mel-Masking, LJ-Speech contamination, 64.8% accuracy							
GradCam	6.50	39.05	3.06	0.33	0.20	0.70	9.66	0.18
GradCam++	12.85	32.81	6.50	**0.47**	0.32	0.41	10.36	0.35
SmoothGrad	0.63	47.40	0.17	0.03	0.02	0.28	10.55	0.04
IG	0.53	47.70	0.10	0.10	0.01	0.56	10.12	0.04
LMAC-ZS (CT)	**24.38**	**20.69**	**11.29**	0.43	**0.56**	0.80	9.26	0.11
LMAC-ZS (Full)	8.95	30.55	3.69	0.38	0.35	**0.86**	**8.79**	0.10
	ZS classification on ESC50, STFT-Masking, LJ-Speech contamination, 64% accuracy							
GradCam	24.93	22.91	**12.78**	**0.67**	0.50	0.70	11.72	0.18
GradCam++	34.13	**12.24**	10.84	**0.67**	**0.72**	0.41	12.44	0.34
SmoothGrad	9.18	29.60	3.91	0.67	0.40	0.53	12.05	0.00
IG	15.55	27.15	6.51	0.66	0.46	**0.73**	11.67	0.01
LMAC-ZS (CT)	**25.77**	17.79	9.67	0.63	0.63	0.61	11.96	0.04
LMAC-ZS (Full)	25.73	15.90	7.23	0.66	0.62	0.72	**11.47**	0.05

reaches better performance in terms of AI, AD, AG, and very comparable numbers in terms of Fid-In. We also observe that in terms of Sparsity and Complexity LMAC-ZS yields better masks in the Mel Domain. In the STFT domain except for LJ-Speech contamination, we observe that LMAC-ZS obtains better performance in terms of AI, AD, and AG. We would like to note that GradCAM++ obtains better FF numbers in general, but we note that GradCAM++ mask areas are larger as shown in the last column with MM. We also observe similar trends for the explanations obtained on US8K samples contaminated with various background noises shown in Table 3. Another point to note is that in general LMAC-ZS trained on the full CLAP training set yields better performance. However, we observe that training LMAC-ZS only on the Clotho dataset yields to comparable or better performance (e.g. ESC50, Mel, white noise contamination). This shows that, in situations where there is limited access to computational resources, training only on Clotho can produce faithful explanations. We furthermore compare the effect of changing the size of the training set size for the interpreter in Appendix C.

4.4 Qualitative Comparison and Sanity Checks

We provide some qualitative examples of generated explanations in Figure 4, and compare with GradCAM++ which seems to provide the most faithful explanations among the baselines according

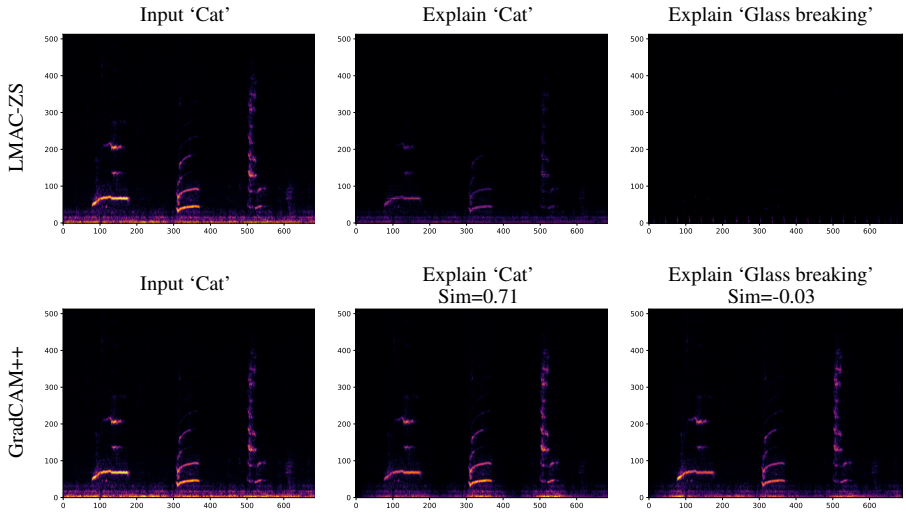

Figure 4: Qualitative Comparisons of Explanations given by LMAC-ZS, and GradCAM++, for two different classes. We see that LMAC-ZS shuts-off the explanation depending on the similarity of the given prompt with the input audio, whereas GradCAM++ remains insensitive to the class label.

to the Tables 1, 2, and 3. We see that LMAC-ZS generates explanations that are much more sensitive to the similarity between the text prompt and the input audio. For instance in LMAC-ZS explanations we see that if there exists a large similarity between the text prompt and the input audio, the mask correctly highlights relevant portions of the input spectrogram. Also, we see that if the similarity between the input and the text prompt is small then the mask tends not to highlight any areas as expected. For instance in Figure 4, we see for the input recordings that corresponds to a ‘Cat’, both LMAC-ZS and GradCAM++ return reasonable explanations. However, when we prompt LMAC-ZS for an unrelated prompt (e.g. ‘Glass Breaking’ in this case), it correctly returns an empty explanation mask, as it is impossible to explain. On the contrary, when GradCAM++ returns a class activation map corresponding to the class "Glass Breaking," we observe that the explanation remains unchanged.

To measure the correlation between the mask mean and similarity, Figure 3 presents a scatter plot depicting the relationship between the similarity of the input text prompt and audio. For LMAC-ZS, we observe that explanations are appropriately returned as empty (indicating small Mask-Means) when the similarity score, estimated using CLAP embeddings, is low. Whereas for GradCAM++, the mask mean and similarity appear to be independent of each other.

Finally, we conduct a cascading model randomization sanity check [18] to assess the sensitivity of explanations returned by LMAC-ZS to the CLAP weights. As illustrated in Figure 3, after three layers of randomization, the similarity drastically decreases for LMAC-ZS, while it remains constant for GradCAM++. We visualize these explanations in Figure 5 and provide additional samples in Appendix A.2. More qualitative samples are available through our companion website[4].

5 Limitations and Societal Impact

Limitations: Our current implementation focuses on fixed-length audio for simplicity. However, the core methodology of LMAC-ZS can be extended to handle variable-length inputs. Additionally, while this work employs standard faithfulness metrics that analyze the dominant class contribution, LMAC-ZS allows for investigating contributions from the top k classes. Studying the top k contributions to faithfulness could provide further insights into the model’s decision-making process. Lastly, our study is limited to the CLAP model, primarily selected for its widespread adoption within the field. It is worth mentioning that there is limited availability of alternatives. For instance, most alternative models such as LAION CLAP [47] are still variations of CLAP, offering minimal differences in their core architecture.

[4] `https://francescopaissan.it/lmaczs`

https://doi.org/10.52202/079017-2087

Societal Impact: We believe this research has the potential for societal benefits, particularly in healthcare applications. While this work does not directly target medical diagnosis, improved explainability of audio classifiers for speech pathologies could make them more trustworthy and accepted by medical professionals. We do not see direct negative societal impacts from this research.

6 Conclusions

This paper, to the best of our knowledge, represents the first attempt to develop a model specifically designed for interpreting the decisions of pre-trained zero-shot audio classifiers. In particular, we introduce LMAC-ZS, a novel post-hoc explanation method employing a specialized decoder that generates saliency maps highlighting the regions of the audio input that most contribute to the model predictions. Extensive evaluations highlighted that LMAC-ZS effectively generates explanations that closely align with the decisions made by the CLAP model in zero-shot settings. Our quantitative and qualitative comparisons show that LMAC-ZS outperforms or is comparable to the most popular baseline saliency methods on most quantitative faithfulness metrics. Additionally, LMAC-ZS offers the possibility of being prompted for an explanation. This ability is missing in traditional methods and allows users to gain further insights into the decision-making processes conducted by the model.

Acknowledgements

This research was enabled in part by support provided by Calcul Québec and the Digital Research Alliance of Canada, and the funds provided by Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] Christoph Molnar. Interpretable machine learning, 2022.

[2] Pablo Zinemanas, Martín Rocamora, Marius Miron, Frederic Font, and Xavier Serra. An interpretable deep learning model for automatic sound classification. *Electronics*, 10, 2021.

[3] Pablo Alonso-Jiménez, Leonardo Pepino, Roser Batlle-Roca, Pablo Zinemanas, Dmitry Bogdanov, Xavier Serra, and Martín Rocamora. Leveraging pre-trained autoencoders for interpretable prototype learning of music audio. In *ICASSP Workshop on Explainable AI for Speech and Audio (XAI-SA)*, 2024.

[4] Luca Della Libera, Cem Subakan, and Mirco Ravanelli. Focal modulation networks for interpretable sound classification. In *ICASSP Workshop on Explainable AI for Speech and Audio (XAI-SA)*, 2024.

[5] Jayneel Parekh, Sanjeel Parekh, Pavlo Mozharovskyi, Florence d'Alché-Buc, and Gaël Richard. Listen to interpret: Post-hoc interpretability for audio networks with NMF. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

[6] Verena Haunschmid, Ethan Manilow, and Gerhard Widmer. audioLIME: Listenable explanations using source separation. In *International Workshop on Machine Learning and Music*, 2020.

[7] Francesco Paissan, Cem Subakan, and Mirco Ravanelli. Posthoc interpretation via quantization. *arXiv preprint arXiv:2303.12659*, 2023.

[8] Francesco Paissan, Mirco Ravanelli, and Cem Subakan. Listenable Maps for Audio Classifiers. In *International Conference on Machine Learning (ICML)*, 2024.

[9] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[10] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. CLAP: Learning audio concepts from natural language supervision. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[11] Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Annual ACM Conference on Multimedia*, 2015.

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

[13] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop Track*, 2014.

[15] 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*, 2019.

[16] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*, 2017.

[17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[19] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[20] Ruth Fong and Andrea Vedaldi. Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[21] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference (BMVC)*, 2018.

[22] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations (ICLR)*, 2019.

[23] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[24] Lijie Fan, Shengjia Zhao, and Stefano Ermon. Adversarial localization network. In *Learning with limited labeled data: weak supervision and beyond, NeurIPS Workshop*, 2017.

[25] Konrad Zolna, Krzysztof J. Geras, and Kyunghyun Cho. Classifier-agnostic saliency map extraction. In *AAAI Conference on Artificial Intelligence*, volume 33, 2019.

[26] Sören Becker, Johanna Vielhaben, Marcel Ackermann, Klaus-Robert Müller, Sebastian Lapuschkin, and Wojciech Samek. AudioMNIST: Exploring explainable artificial intelligence for audio analysis on a simple benchmark. *Journal of the Franklin Institute*, 361, 2024.

[27] Hannah Muckenhirn, Vinayak Abrol, Mathew Magimai-Doss, and Sébastien Marcel. Understanding and Visualizing Raw Waveform-Based CNNs. In *Proc. Interspeech*, 2019.

[28] Saumitra Mishra, Bob L. Sturm, and Simon Dixon. Local interpretable model-agnostic explanations for music content analysis. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[29] Saumitra Mishra, Emmanouil Benetos, Bob L. Sturm, and Simon Dixon. Reliable local explanations for machine listening. In *International Joint Conference on Neural Networks (IJCNN)*, 2020.

 https://doi.org/10.52202/079017-2087

[30] Shreyan Chowdhury, Verena Praher, and Gerhard Widmer. Tracing back music emotion predictions to sound sources and intuitive perceptual qualities. In *Sound and Music Computing Conference (SMC)*, 2021.

[31] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 1999.

[32] Aaron Van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[33] Hanwei Zhang, Felipe Torres, Ronan Sicre, Yannis Avrithis, and Stephane Ayache. Opti-CAM: Optimizing saliency maps for interpretability. *arXiv preprint arXiv:2301.07002*, 2023.

[34] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning (ICML)*, volume 119, 2020.

[35] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34), 2023.

[36] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[37] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[38] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2022.

[39] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[40] Irene Martín-Morató and Annamaria Mesaros. What is the ground truth? reliability of multi-annotator data for audio tagging. In *European Signal Processing Conference (EUSIPCO)*, 2021.

[41] Keith Ito and Linda Johnson. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[43] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[44] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[45] Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain de Langen, Cem Subakan, Peter Plantinga, Yingzhi Wang, Pooneh Mousavi, Luca Della Libera, Artem Ploujnikov, Francesco Paissan, Davide Borra, Salah Zaiem, Zeyu Zhao, Shucong Zhang, Georgios Karakasidis, Sung-Lin Yeh, Pierre Champion, Aku Rouhe, Rudolf Braun, Florian Mai, Juan Zuluaga-Gomez, Seyed Mahed Mousavi, Andreas Nautsch, Xuechen Liu, Sangeet Sagar, Jarod Duret, Salima Mdhaffar, Gaelle Laperriere, Mickael Rouvier, Renato De Mori, and Yannick Esteve. Open-source conversational ai with SpeechBrain 1.0, 2024.

[46] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

[47] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

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

[49] Peijie Chen, Qi Li, Saad Biaz, Trung Bui, and Anh Nguyen. gscorecam: What objects is clip looking at? In *Computer Vision – ACCV 2022: 16th Asian Conference on Computer Vision, Macao, China, December 4–8, 2022, Proceedings, Part IV*, page 588–604, Berlin, Heidelberg, 2023. Springer-Verlag.

 https://doi.org/10.52202/079017-2087

A Appendix / supplemental material

A.1 Results on UrbanSound8K Dataset with Contaminations

Table 3: Out-of-Domain quantitative evaluation for the UrbanSound8K Dataset.

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	ZS classification on US8K, Mel-Masking, US8K contamination, 57% accuracy							
GradCam	2.64	48.43	1.43	0.27	0.12	0.77	9.42	0.13
GradCam++	7.58	37.89	3.91	**0.56**	0.33	0.37	10.39	0.40
SmoothGrad	2.16	50.12	1.14	0.05	0.08	0.32	10.51	0.04
IG	1.82	49.79	0.82	0.18	0.07	0.59	10.06	0.03
LMAC-ZS (CT)	17.74	25.57	9.87	0.48	0.55	**0.86**	**8.95**	0.07
LMAC-ZS (Full)	**36.08**	**16.98**	**19.23**	0.47	**0.69**	0.77	9.00	0.19
	ZS classification on US8K, STFT-Masking, ESC50 contamination, 57% Accuracy							
GradCam	17.83	31.78	12.05	0.78	0.42	0.76	11.51	0.13
GradCam++	28.81	14.56	14.42	0.78	0.73	0.37	12.48	0.39
SmoothGrad	23.13	20.58	13.73	**0.79**	0.64	0.52	12.12	0.0002
IG	21.53	22.41	12.76	0.74	0.60	0.77	11.53	0.01
LMAC-ZS (CT)	31.09	17.69	15.29	0.72	0.66	0.55	12.12	0.08
LMAC-ZS (Full)	**39.42**	**11.53**	**17.51**	0.75	**0.78**	**0.78**	**11.23**	0.06
	ZS classification on US8K, Mel-Masking, White Noise contamination, 62% accuracy							
GradCam	6.77	44.01	3.91	0.35	0.21	0.73	9.46	0.16
GradCam++	12.51	37.77	8.49	**0.60**	0.31	0.38	10.38	0.39
SmoothGrad	3.55	49.01	1.60	0.04	0.11	0.31	10.52	0.03
IG	2.51	48.43	0.94	0.08	0.13	0.56	10.11	0.03
LMAC-ZS (CT)	**42.70**	**12.02**	**25.78**	0.42	0.76	0.87	8.91	0.07
LMAC-ZS (Full)	34.53	14.13	20.32	0.39	**0.80**	**0.88**	**8.72**	0.08
	ZS classification on US8K, STFT-Masking, White Noise contamination, 61.1% accuracy							
GradCam	18.24	35.12	12.24	**0.76**	0.34	**0.74**	**11.48**	0.15
GradCam++	20.16	27.33	13.21	**0.76**	0.49	0.38	12.48	0.38
SmoothGrad	21.36	27.98	14.25	**0.76**	0.47	0.52	12.21	0.0004
IG	19.91	33.36	13.74	0.72	0.36	0.69	11.79	0.01
LMAC-ZS (CT)	27.78	17.64	13.44	0.69	0.66	0.59	12.05	0.07
LMAC-ZS (Full)	**46.51**	**9.95**	**25.28**	0.69	**0.81**	0.70	11.60	0.06
	ZS classification on US8K, Mel-Masking, LJ-Speech contamination, 44.9% accuracy							
GradCam	3.49	46.48	1.69	0.28	0.14	0.68	9.68	0.19
GradCam++	10.86	36.61	6.28	**0.45**	0.32	0.37	10.39	0.41
SmoothGrad	2.04	50.09	1.10	0.03	0.05	0.31	10.35	0.04
IG	1.69	49.80	0.74	0.12	0.05	0.60	10.03	0.03
LMAC-ZS (CT)	25.78	23.54	17.43	0.37	0.55	0.86	8.93	0.07
LMAC-ZS (Full)	**36.24**	**13.90**	**20.47**	0.41	**0.73**	**0.86**	**8.79**	0.10
	ZS classification on US8K, STFT-Masking, LJ-Speech contamination, 46.1% accuracy							
GradCam	21.48	28.71	14.13	**0.76**	0.45	0.69	11.74	0.19
GradCam++	**38.74**	**11.53**	17.95	**0.76**	**0.76**	0.37	12.47	0.40
SmoothGrad	34.35	19.43	24.32	**0.76**	0.62	0.52	12.11	0.00
IG	34.57	20.43	**26.10**	0.69	0.60	0.74	11.59	0.01
LMAC-ZS (CT)	35.96	15.91	18.33	0.68	0.67	0.63	11.92	0.07
LMAC-ZS (Full)	32.51	13.79	15.77	0.72	0.74	**0.79**	**10.99**	0.02

A.2 Qualitative Analysis of Model Randomization Test

Figure 5 presents a qualitative visualization of Model Randomization Test results for GradCAM++ and LMAC-ZS.

A.3 Qualitative Comparison with GradCAM++

Figures 6, 7 show an additional sample for the quality of the explanations on spectra.

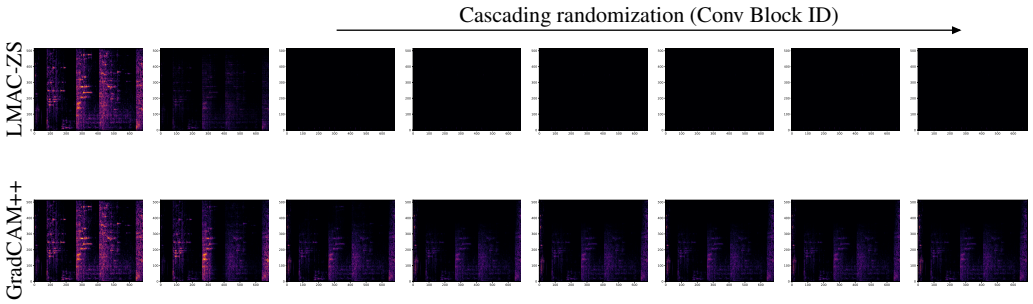

Figure 5: Visualization of Explanations after Cascading Model Randomization. Left column is the input, second column is the original explanation, and more we go towards the right more layers are randomized. Top row is for LMAC-ZS, and the bottom row is for GradCAM++.

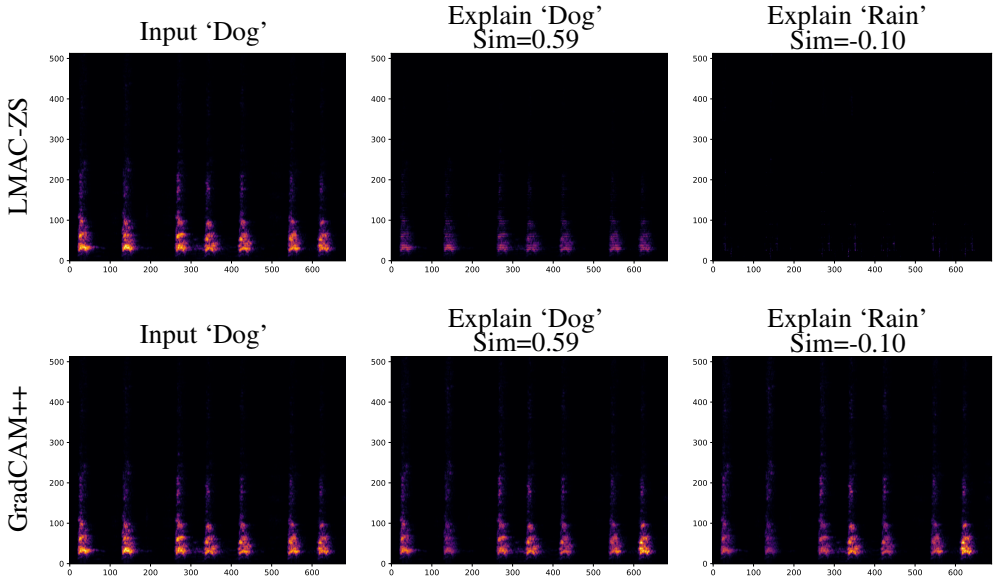

Figure 6: Qualitative Comparisons

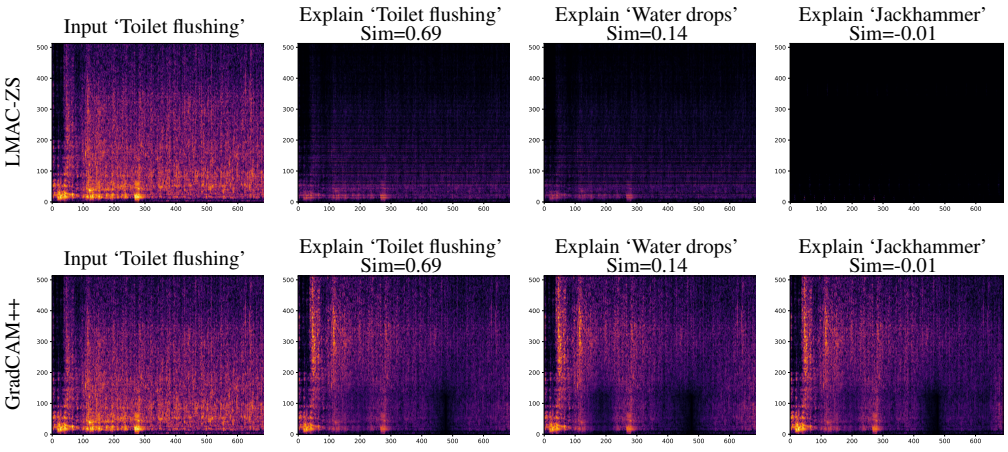

Figure 7: Qualitative Comparisons 2

https://doi.org/10.52202/079017-2087

B Explanation sensitivity to Audio-Text Similarity

To showcase the effectiveness of the additional diversity term (Equation 7), we conducted qualitative and quantitative tests to evaluate the explanation sensitivity to text prompts.

B.1 Qualitative results

We compare the explanations obtained with LMAC-ZS with and without the diversity term in Eq. 7. We present the results in Figure 8 and Figure 9, respectively. The results are obtained for the model that does masking in the STFT domain and was trained on FSD50K. We present the plots using log-frequency scaling. In each plot, we give the original text-audio similarity (in the title of the first subplot), as well as the audio-text similarity after masking the audio (in the title of the third subplot). Note that the predicted class is also used as the prompt for the masking model. We observe that masks are more sensitive to text prompts because of the additional diversity term.

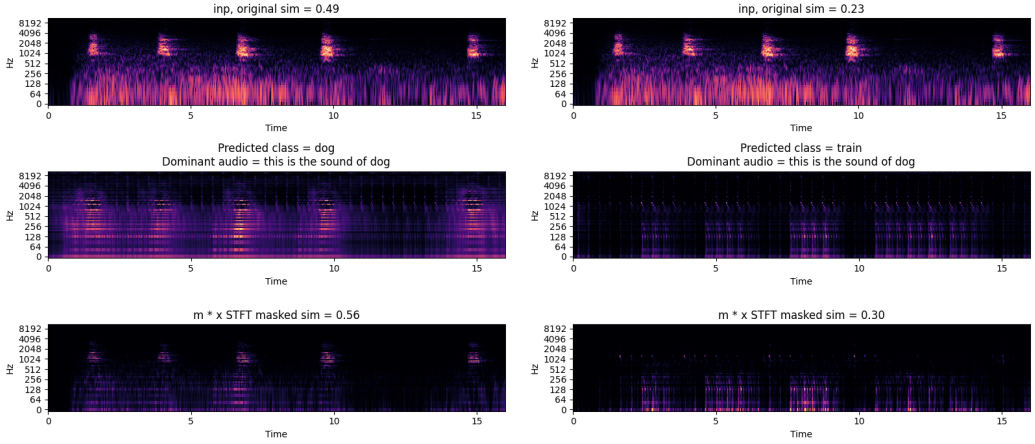

Figure 8: Explanations obtained with the additional diversity term (Eq 7).

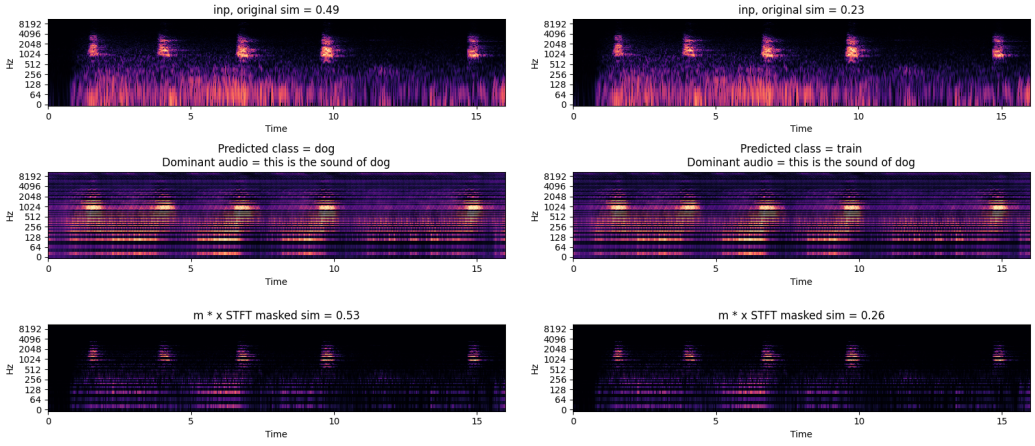

Figure 9: Explanations obtained without the additional diversity term (Eq 7).

B.2 Quantitative results

In Figure 10, we present the 2D histogram of mask mean and similarity between text and audio after masking. This highlights the increased mask sensitivity of the model to different text prompts when the diversity term in Equation 7 of the paper is utilized. We see in the left panel of Fig. 10 that without the diversity term, the mask means do not have a discernible correlation with the text-masked audio similarity.

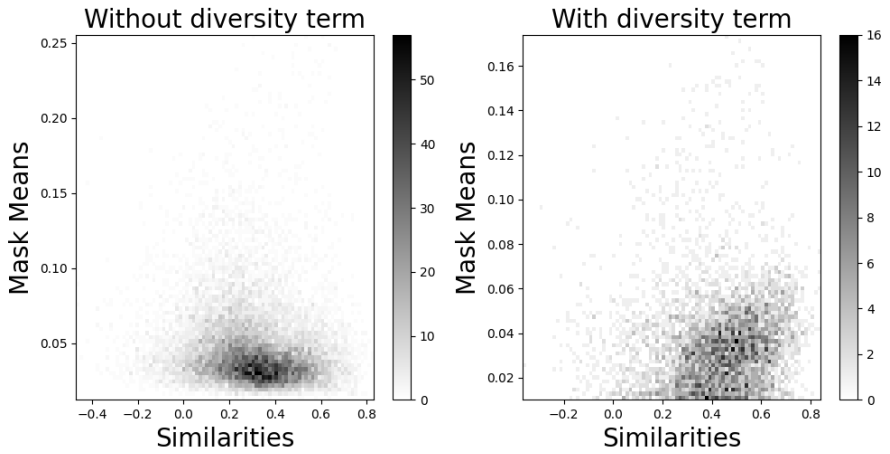

Figure 10: Audio-text similarity after audio masking, without the diversity term (left), with the diversity term (right).

C Ablation on the training dataset size for LMAC-ZS

Table 4: Interpreter performance for different training dataset sizes and for additional baselines.

Metric	AI (↑)	AD (↓)	AG (↑)	FF (↑)	Fid-In (↑)	SPS (↑)	COMP (↓)	MM
	ZS classification on ESC50, STFT-Masking, 80.7% accuracy							
ScoreCAM	29.97	12.14	8.82	0.70	0.75	0.32	12.59	0.41
GScoreCAM	29.64	8.56	6.62	**0.79**	0.84	0.36	12.52	0.39
LMAC-ZS Clotho	37.40	7.43	**11.26**	0.78	0.86	0.50	12.29	0.11
LMAC-ZS FSD50K	34.00	8.33	10.12	0.77	0.83	0.61	11.83	0.04
LMAC-ZS AudioCaps	39.00	5.93	10.43	0.78	0.88	**0.68**	**11.67**	0.07
LMAC-ZS MACs	15.61	22.86	5.32	0.78	0.61	0.42	12.42	0.04
LMAC-ZS Subset (25%)	41.50	**3.48**	7.99	**0.79**	**0.92**	0.65	11.91	0.22
LMAC-ZS Subset (50%)	**43.70**	3.54	7.86	**0.79**	0.91	0.63	11.97	0.19
LMAC-ZS Subset (75%)	40.60	4.74	7.73	**0.79**	0.89	0.66	11.84	0.17
LMAC-ZS All Data	43.35	4.29	10.57	0.78	0.90	0.65	11.86	0.10
	ZS classification on ESC50, STFT-Masking, ESC50 contamination, 58.6% accuracy							
ScoreCAM	31.39	7.03	7.05	**0.79**	**0.87**	0.36	12.52	0.39
GScoreCAM	28.07	13.74	8.42	0.70	0.73	0.32	12.59	0.41
LMAC-ZS Clotho	35.65	12.23	**13.04**	0.69	0.74	0.53	12.18	0.09
LMAC-ZS AudioCaps	35.97	10.35	11.42	0.68	0.76	0.71	11.63	0.07
LMAC-ZS FSD50K	26.95	16.26	9.97	0.67	0.65	0.66	**11.59**	0.03
LMAC-ZS MACS	11.38	31.54	4.42	0.68	0.38	0.44	12.41	0.05
LMAC-ZS Subset (25%)	**42.65**	**5.99**	9.81	0.70	0.84	0.66	11.90	0.20
LMAC-ZS Subset (50%)	39.47	7.52	9.05	0.71	0.81	0.66	11.88	0.16
LMAC-ZS Subset (75%)	40.42	7.07	8.84	0.70	0.83	**0.68**	11.80	0.16
LMAC-ZS All Data	39.47	8.28	11.81	0.69	0.80	0.67	11.79	0.09

LMAC-ZS is a decoder-based interpreter. That is, we train the decoder based on the pre-trained classifier's representations. The amount and quality of training data can impact both the performance and the training time of the interpreter. We benchmarked our interpreter on different datasets with different sizes, i.e. the datasets that are included within the whole CLAP training set (denoted with All Data in Table 4) - The datasets that make up the whole CLAP training set are, Clotho [37], MACs [40], FSD50k [38] and AudioCaps [39]. We have also experimented with randomly subsampling the whole CLAP training set and denoted it as 'Subset' in 4.

In Table 4, we report the interpreter performance for the aforementioned training datasets with different sizes. We note that the explanation's faithfulness is comparable when training the decoder on the entire training data or a subset; this indicates that it is possible to train LMAC-ZS on a smaller dataset and still obtain faithful explanations.

https://doi.org/10.52202/079017-2087

Table 5: Frechet Audio Distance of the training datasets with respect to ESC-50.

	MACs	FSD50k	Clotho	AudioCaps	Subset (25%)
ESC-50	3.33	3.04	3.09	3.11	3.18

However, we note that the MACs-only training results obtained the lowest performance on ESC50. We note that this is likely related to the differences in data distributions. To investigate this, we have computed Frechet Audio Distances (computed via CLAP embeddings) between ESC-50 and different subsets in Table 5. We observe that the highest distance is between MACs and ESC50 is the highest. This suggests that if the similarity between the target domain and the training set for the interpreter is relatively high, it is possible to train LMAC-ZS on smaller subsets.

In Table 4, we also present the performance of two additional baselines, ScoreCAM [48] and GScoreCAM [49], and we see that on ESC50, except the case where input audio is contaminated with another audio sample from the ESC50 dataset, LMAC-ZS is able outperform these baselines for the majority of the faithfulness metrics.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   [Yes]

   Justification: The abstract and introduction clearly outline the development of LMAC-ZS as our primary contribution. This claim is supported by a detailed description of LMAC-ZS, its training process, and evaluation methodology in the subsequent sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the study are clearly described in Section 5 (Limitations and Societal Impact). Additionally, the paper addresses the computational resources required for the experiments in Section 4.2 (Experimental Setup).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

https://doi.org/10.52202/079017-2087

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides a detailed description of the experimental setup in Section 4.2. Additionally, to ensure reproducibility, the code will be publicly released using a popular toolkit like SpeechBrain. The code repository also includes documentation on how to run the experiments and replicate the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code will be made publicly available. We provide documentation on how to run the experiments and replicate the results discussed in this paper. Our results are fully replicable as we only used publicly available datasets.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 details the experimental setup, including the datasets employed and the evaluation metrics used. More granular implementation details are available in the code repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We think that the absence of error bars or explicit statistical significance measures does not detract from the robustness of our findings. The substantial performance gap observed between our proposed method and existing approaches gives us a high level of confidence in the reliability and significance of our results.

https://doi.org/10.52202/079017-2087

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.2 details the computational resources employed in our experiments. This includes information on the type of GPU, memory requirements, and training times.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and believe our research is fully compliant with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 5 (Limitations and Societal impact) discusses it.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

https://doi.org/10.52202/079017-2087

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides a detailed description of LMAC-ZS, including its architecture, training process, and evaluation methodology (Sections 3 and 4). Additionally, the code for LMAC-ZS will be released publicly on SpeechBrain alongside comprehensive documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-2087