

Masked Spectrogram Modeling using Masked Autoencoders for Learning General-purpose Audio Representation

Daisuke Niizumi

DAISUKE.NIIZUMI.DT@HCO.NTT.CO.JP

Daiki Takeuchi

DAIKI.TAKEUCHI.UX@HCO.NTT.CO.JP

Yasunori Ohishi

YASUNORI.OOISHI.UK@HCO.NTT.CO.JP

Noboru Harada

NOBORU.HARADA.PV@HCO.NTT.CO.JP

Kunio Kashino

KUNIO.KASHINO.ME@HCO.NTT.CO.JP

NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan

Editors: Joseph Turian, Björn W. Schuller, Dorien Herremans, Katrin Kirchhoff, Paola Garcia Perera, and Philippe Esling

Abstract

Recent general-purpose audio representations show state-of-the-art performance on various audio tasks. These representations are pre-trained by self-supervised learning methods that create training signals from the input. For example, typical audio contrastive learning uses temporal relationships among input sounds to create training signals, whereas some methods use a difference among input views created by data augmentations. However, these training signals do not provide information derived from the intact input sound, which we think is suboptimal for learning representation that describes the input as it is.

In this paper, we seek to learn audio representations from the input itself as supervision using a pretext task of auto-encoding of masked spectrogram patches, Masked Spectrogram Modeling (MSM, a variant of Masked Image Modeling applied to audio spectrogram). To implement MSM, we use Masked Autoencoders (MAE), an image self-supervised learning method. MAE learns to efficiently encode the small number of visible patches into latent representations to carry essential information for reconstructing a large number of masked patches. While training, MAE minimizes the reconstruction error, which uses the input as training signal, consequently achieving our goal.

We conducted experiments on our MSM using MAE (MSM-MAE) models under the evaluation benchmark of the HEAR 2021 NeurIPS Challenge. Our MSM-MAE models outperformed the HEAR 2021 Challenge results on seven out of 15 tasks (e.g., accuracies of 73.4% on CREMA-D and 85.8% on LibriCount), while showing top performance on other tasks where specialized models perform better. We also investigate how the design choices of MSM-MAE impact the performance and conduct qualitative analysis of visualization outcomes to gain an understanding of learned representations. We have made our code available online for further improvements and applications of the MSM framework.¹

Keywords: Self-supervised learning, General-purpose Audio Representation, Masked Autoencoders, Masked Spectrogram Modeling

1. Introduction

With the recent progress of audio representation learning, general-purpose audio representations have shown good performance in various audio tasks (Saeed et al., 2021; Niizumi et al., 2021; Wang et al., 2022). While previous supervised learning methods (Hershey et al.,

1. <https://github.com/nttcslab/msm-mae>

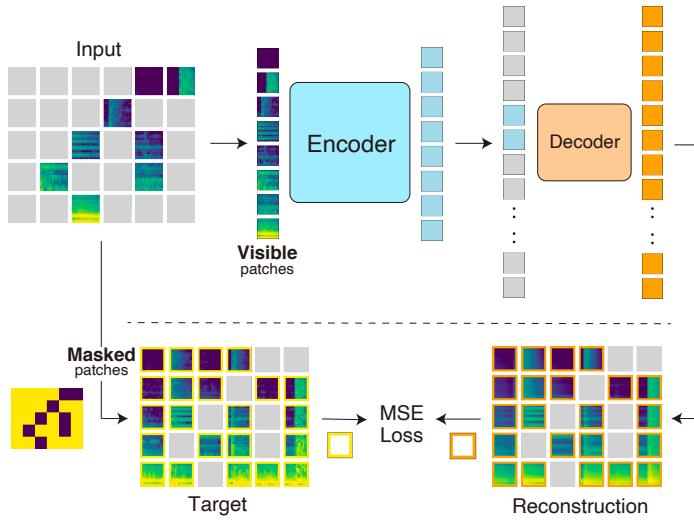


Figure 1: MAE (He et al., 2022) pre-training flow; we redraw Figure 1 in the MAE paper, in which we replaced the input image with a spectrogram and added loss calculation flow. MAE masks 75% of input patches, then the encoder processes the visible 25% of patches only, saving 75% of computation load. The lightweight decoder takes as input the encoded 25% plus mask tokens that fill the masked 75% of the input, then reconstructs spectrogram. The loss calculates the reconstruction error of the masked patches, which is a mean squared error (MSE).

2017; Kong et al., 2020; Koutini et al., 2022) learn to discriminate labels, these general-purpose audio representations are pre-trained by self-supervised learning methods that do not rely on labels.

These methods utilize self-supervising signals such as the temporal relationship between audio samples or differences in multiple audio samples generated by data augmentations. For example, triplet loss or contrastive learning methods (Shor et al., 2020; Saeed et al., 2021; Spijkervet and Burgoyne, 2021; Fonseca et al., 2021) learn to make representations closer to temporally close audio segments while pushing away remote segments. Our previous study BYOL-A (Niizumi et al., 2021) learns representations invariant against the difference in audio signals created by data augmentations.

However, training signals of these methods do not provide information about *complete details* of the input. The temporal relationship among audios can be the information about the difference between the audio signals that do not describe input details, and the data augmentations can change the details in original input audio signals. Therefore, we think these training signals are suboptimal for learning to represent audio input as it is.

We think that the input signal itself can be the best training signal to learn representations that describe the input in detail. Learning frameworks to achieve our goal include Masked Language Modeling (MLM) in natural language processing (NLP) or Masked Image Modeling (MIM) in the image domain. These methods learn representations by masking a part of the input signal and using other parts to predict the masked signals. In partic-

ular, MLM such as BERT (Devlin et al., 2019) have already proven highly effective and have demonstrated strong performance. Inspired by BERT, speech self-supervised learning methods that learn from masked input prediction (Baevski et al., 2020; Liu et al., 2020; Chi et al., 2021; Hsu et al., 2021), have also shown solid results in the audio domain. In the image domain, recent progress of MIM such as BEiT (Bao et al., 2022) and Masked Autoencoders (MAE) (He et al., 2022) have shown promising performance such that “*Self-supervised learning in vision may now be embarking on a similar trajectory as in NLP*” (He et al., 2022).

In this study, we explore the learning of general-purpose audio representations through the MIM applied to the audio spectrogram, which we call Masked Spectrogram Modeling (MSM). MIM splits the input image into grid patches. Therefore, applying MIM to the audio spectrogram splits the input along the time and frequency axes, allowing “*the model to learn both the temporal and frequency structure*” (Gong et al., 2022), unlike previous methods for speech (e.g., Mockingjay (Liu et al., 2020) and wav2vec 2.0 (Baevski et al., 2020)) that split audio along time only.

To implement MSM, we use MAE as a training framework. MAE learns to efficiently encode the small number of visible patches into latent representations to carry essential information for reconstructing masked patches, a large portion of the input signal. It then calculates the reconstruction error as a training loss, achieving our goal of using the input itself as a training signal.

Our main contributions are the proposal and implementation of MSM using MAE (MSM-MAE) to learn general-purpose audio representations and results showing that our variants of MAE outperform other methods on some tasks in the HEAR 2021 NeurIPS Challenge (Turian et al., 2022). In addition, we investigate how the design choices of MSM-MAE impact the performance and present qualitative analyses of learned representations using visualizations. Our code is available online at <https://github.com/nttcslab/msm-mae>.

2. Related Work

Audio representation learning closely related to our work. SSAST (Gong et al., 2022) is a self-supervised learning method that pre-trains ViT (Dosovitskiy et al., 2021) using a pretext task of *Joint Discriminative and Generative Masked Spectrogram Patch Modeling (MSPM)*, which combines contrastive learning and masked patch reconstruction. While the patch reconstruction task is the same as that with MAE, it uses a two-layer MLP to reconstruct masked patches, unlike MAE, which uses a sufficiently deep transformer.

Mockingjay (Liu et al., 2020) proposed *Masked Acoustic Model (MAM)*, a pretext task of reconstructing the masked time frames; the subsequent studies TERA (Liu et al., 2021) and Audio ALBERT (Chi et al., 2021) also use MAM. Unlike MIM, MAM slices the spectrogram along time as natural handling of time-series data, samely as the methods that accept raw audio as input, such as wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021).

PaSST (Koutini et al., 2022) is a supervised learning method that pre-trains ViT with the proposed *Patchout*, taking a similar approach to MAE. The *Patchout* reduces the number of patches to encode, thus, saving computation resources.

Audio self-supervised learning. Methods based on triplet (Shor et al., 2020) or contrastive loss (Saeed et al., 2021; Fonseca et al., 2021; Spijkervet and Burgoyne, 2021; Wang

(et al., 2022) learn from the temporal relationship between audios; they learn to make feature embeddings of audios closer for the audios temporally closer, or push them away for the ones temporally remote. Then, these methods do not model the representation of the input audio as it is.

On the other hand, our previous study BYOL-A (Niizumi et al., 2021) and SERAB BYOL-S (Scheidwasser-Clow et al., 2022) use data augmentations to produce multiple audios with a small degree of difference from the same audio input; they learn to make invariant representations for these augmented audios. Therefore, these methods may not learn part of the audio in which data augmentations make changes.

Masked Image Modeling. ViT (Dosovitskiy et al., 2021) conducts a self-supervised learning of predicting average 3bit color of masked patch as a preliminary exploration, resulted in 4% behind supervised pre-training version of the ViT. BEiT (Bao et al., 2022) proposes a masked image modeling task that learns to predict discrete visual tokens of masked patches. BEiT outperforms supervised pre-trained ViT; however, it pre-trains the model to encode the input image into discrete tokens rather than representing the input as it is, and requires pre-trained discrete VAE (Ramesh et al., 2021), which is not available for audio.

2.1. Masked Autoencoders (MAE)

MAE (He et al., 2022) reconstructs the original signal given its partial observation. Figure 1 illustrates the pre-training flow with a spectrogram as input. First, MAE splits the input into patches, adds positional encodings, and randomly masks a large part of the patches. Then an encoder (a ViT) processes the visible patches only to latent representations. Next, a decoder (a smaller transformer) reconstructs input from the latent representations of visible patches and mask tokens representing masked patches. Then the loss is calculated as the mean squared error (MSE) for all the masked patches between the reconstruction and target, which is the normalized input. After pre-training, only the encoder is applied, and it encodes whole patches of input images to produce representations for downstream tasks.

MAE masks a very large portion (e.g., 75%) of patches with a notion that information density is different from that in languages and images; images are natural signals with spatial redundancy compared to languages, which are highly semantic and information-dense. The MAE paper shows that the optimal mask ratio is 75%, much higher than the 15% of BERT (Devlin et al., 2019) in the NLP domain.

Unlike classical autoencoders, MAE has an asymmetric encoder-decoder design; an encoder operates on the partial visible signal only, whereas a lightweight decoder reconstructs the full signal. These design choices save computation load and enable us to scale MAE to train large models efficiently.

A sufficiently deep decoder is essential for linear evaluation performance without fine-tuning. The last several layers in an autoencoder can be more specialized for reconstruction, thus becoming less relevant for other tasks. A reasonably deep decoder can help make latent representations from the encoder output more abstract (He et al., 2022; Cao et al., 2022).

Concurrent works that apply MAE to audio include MAE-AST (Baade et al., 2022), MaskSpec (Chong et al., 2022), and Audio-MAE (Huang et al., 2022). MAE-AST is based on SSAST (Gong et al., 2022) architecture and uses a shallow two-layer transformer as the decoder, similar to ours. MaskSpec uses the same encoder as PaSST (Koutini et al., 2022), and Audio-MAE uses the *local attention mechanism* with its decoder. These models are evaluated by *fine-tuning*, making the experimental results incompatible with the HEAR benchmark results that evaluate the *frozen* pre-trained models. Unlike these methods, our MSM-MAE is based on vanilla MAE with a smaller four-layer decoder and evaluated on the HEAR benchmark.

3. Masked Spectrogram Modeling using Masked Autoencoders

We apply Masked Image Modeling to the audio spectrogram, which we call Masked Spectrogram Modeling (MSM). MSM splits the input along the time and frequency axes, allowing it to learn both the temporal and frequency structure, unlike the previous methods for speech that split along time only. To implement MSM, we use Masked Autoencoders (MSM-MAE). In preliminary experiments, we found that MSM-MAE can follow the basic design choices of the original MAE, except the half decoder depth. While the same mask ratio of 75% suggests that information density is close to that of the image, the half decoder depth might suggest lower complexity of the context of the spectrogram than that of the image.

Besides the original designs, taking spectrogram as input introduces new choices, namely, input and patch size, because spectrogram has different axes of frequency and time, unlike an image. For the input size, which consists of the number of frequency bins and time frames (denoted F and T), we handle a constant F and focus on exploring the optimal T . We confirmed that both T as input audio duration and patch size choices positively influence downstream task performance in preliminary experiments, unlike many other parameters that degrade performance with change. Therefore, we investigate these two design choices in this paper.

In addition, to better use learned features in the downstream tasks, we also introduce a feature calculation specialized to the spectrogram input.

3.1. Input Audio Duration

During the pre-training, the longer duration, the model gains more chance to learn the relationships among the contents. Therefore, the longer duration could result in better representations. Meanwhile, the duration of samples in the downstream tasks ranges from 1-s to tens of seconds, or even longer. In addition, it may be fixed or vary from sample to sample. The optimal duration can depend on the task.

From the perspective of complexity, the shorter duration reduces the computation load of the two transformers on MAE because the length of the input sequence requires quadratic computational and memory complexity on the transformers; thus, the shorter duration is beneficial for scaling the system. For these reasons, we study various input audio durations.

3.2. Patch Size

While both the input and patch sizes are square with the image, we handle rectangle input with the spectrogram. Thus, the same goes for the patch sizes.

The patch size also affects task performance because it sets the frequency/time resolution of encoded representations. There are various task demands; for example, pitch detection requires sufficient frequency resolution, whereas short event detection requires fine time resolution. To meet these demands, we can make the patch smaller to make the resolution finer.

However, the available choice of resolution is limited due to computational complexity; for example, making the patch size half on both frequency and time will quadruple the sequence length. We explore various patch sizes based on the default resolution of 16×16 .

3.3. Feature Calculation for Downstream Tasks

The encoder of the learned MAE encodes whole patches of the audio samples in the task, yielding embeddings of all patches for each audio sample; thereby, the embeddings for a single time frame consist of that of multiple patches of frequencies.

While typically, the patch embeddings for a time frame can be averaged to get a single embedding, we think it impairs available information by averaging embeddings among frequency bins. Therefore, we calculate features by concatenating all the patch embeddings of the same time frame, preserving all available features as the following python pseudo code:

$$\begin{aligned} z' = & z.\text{reshape}(B, N_F, N_T, D) \\ & .\text{transpose}(1, 2) \\ & .\text{reshape}(B, N_T, N_F D) \end{aligned} \tag{1}$$

where $z \in R^{B \times N_F N_T \times D}$ is the encoder output, B is batch size, N_F is the number of patches along frequency, N_T is the number of patches along time, D is a feature dimension, and $z' \in R^{B \times N_T \times N_F D}$ is the calculation result. This calculation summarizes encoded features of a time frame for all frequencies into a single vector. For example, the feature dimension of z' will be 3840 when $D = 768$ and $N_F = 5$, which is used in our experiments.

4. Experiments

We evaluate our MSM-MAE models on a benchmark suite from the HEAR (Holistic Evaluation of Audio Representations) 2021 NeurIPS Challenge ([Turian et al., 2022](#)), which spans multiple audio domains, including speech, environmental sound, and music.

We describe the details of experiments in Section 4.1, and the downstream tasks in Section 4.2. Next, we evaluate our models with results on the HEAR 2021 Challenge in Section 4.3, and investigate how design choices impact the performance in Section 4.4. Then, we analyze learned representations qualitatively using visualizations in Section 4.5.

4.1. Experimental Details

4.1.1. PRE-TRAINING

We used ViT-base (Dosovitskiy et al., 2021) as an encoder model and a smaller decoder with a width of 384-d, depth of 4, and 6 heads. Then, we pre-trained on MAEs with the original parameters except for the pre-training epoch of 100, warmup epoch of 10, batch size of 512, and learning rate of 6e-4. While we normalized batch inputs with dataset statistics, we did not normalize the target when calculating reconstruction loss due to early observation of better performances. We followed the original mask ratio of 0.75.

The pre-training dataset consisted of 1,963,807 samples from balanced_train_segments and unbalanced_train_segments data splits of the AudioSet (Gemmeke et al., 2017). We preprocessed samples to a log-scaled mel spectrogram with a sampling frequency of 16,000 Hz, window size of 25 ms, hop size of 10 ms, and mel-spaced frequency bins $F = 80$ in the range 50–8,000 Hz.

4.1.2. EVALUATION

We evaluated models with input audio duration $T \in \{96, 208, 304, 400, 512\}$, corresponding to 960 ms to 5.12 seconds. We also evaluated models with patch sizes ($F \times T$) of 16×16 by default, 16×8 or 16×4 for double or quadruple time resolutions, and 8×16 for a double frequency resolution. In addition, we evaluated a model with a patch size of 80×4 , which cuts the input along time, making spectrogram strips; note that we use the fixed number of frequency bins $F = 80$. The model configurations are listed in Table 1.

Table 1: Model configuration details.

Model	# of Patches			Patch Size	
	Total	Freq.	Time	Freq.	Time
MSM-MAE-96	30	5	6	16	16
MSM-MAE-208	65	5	13	16	16
MSM-MAE-304	95	5	19	16	16
MSM-MAE-400	125	5	25	16	16
MSM-MAE-512	160	5	32	16	16
MSM-MAE-200 (16×8)	125	5	25	16	8
MSM-MAE-200 (16×4)	250	5	50	16	4
MSM-MAE-208 (8×16)	130	10	13	8	16
MSM-MAE-304 (80×4)	76	1	76	80	4

We used the *hear-eval-kit*² from the HEAR 2021 Challenge. The *hear-eval-kit* evaluates the performance of the models on the downstream tasks without fine-tuning. First, it encodes all the task samples into embeddings using the model as a feature extractor. Then, it trains a shallow downstream model to solve the task, taking the embeddings as input. It reports the test performance of the downstream model as the model performance.

The *hear-eval-kit* requires two types of embeddings from models. One is *timestamp embeddings*, for which we used the z' , features calculated for the downstream task described in Section 3.3; the other is *scene embeddings*, for which we calculated the temporal average

2. <https://github.com/neuralaudio/hear-eval-kit>

of z' . Since the ViT model accepts the fixed input duration T , we convert the variable-length inputs into feature vectors in two steps: encode all the divided segments of length T of input, and then concatenate the encoded features along time. All other details of downstream task evaluation follow the defaults of the *hear-eval-kit*, including the network design of the shallow downstream models for each task.

4.2. Downstream Tasks

We used 15 downstream tasks³ from the HEAR 2021 (Turian et al., 2022), consisting of four environmental sound tasks, five speech tasks, and six music tasks.

Speech tasks. These tasks are for non-semantic speeches, which do not include automatic speech recognition: Speech Commands (Warden, 2018) (SPC, speech command word classification), CREMA-D (Cao et al., 2014) (CRM-D, speech emotion recognition), LibriCount (Stöter et al., 2019) (LbCount, speaker count estimation), Vocal Imitations (Kim et al., 2018a) (VoImit, matching a vocal imitation with an original sound as a classification), and Vox Lingua Top 10 (Valk and Alumäe, 2021) (Lingua10, language identification).

Environmental sound tasks. ESC-50 (Piczak, 2015) (environmental sound classification), FSD50K (Fonseca et al., 2022) (multilabel sound event classification), Gunshot Triangulation (Cooper and Shaw, 2020) (Gunshot, recording location classification), Beehive States (Nolasco et al., 2019) (Beehive, normal or queen-less binary classification).

Music tasks. GTZAN (Tzanetakis and Cook, 2002) (music genre recognition), GTZAN Music Speech (GTZ-M/S, music or speech binary classification), NSynth (Engel et al., 2017) Pitch (NSPitch, pitch classification), Mridingham Stroke and Tonic (Anantapadmanabhan et al., 2013) (Mrd-Stk for stroke, or Mrd-Ton for tonic, pitched percussion stroke or tonic classification), and Beijing Opera Percussion (Tian et al., 2014) (Beijing, percussion instrument classification).

For these tasks, the FSD50K and VoImit results are mAP, the Beehive results are AUCROC, and all other results are accuracies.

Table 2: Environmental sound task results.

Model	Gunshot	FSD50K	ESC-50	Beehive
CREPE (Kim et al., 2018b)	86.3	15.9	30.0	59.3
wav2vec2 (Baevski et al., 2020)	84.8	11.6	56.1	N/A
KW-MLP (Morshed and Ahsan, 2021)	93.2	18.7	36.7	76.0
SERAB BYOL-S (Scheidwasser-Clow et al., 2022)	85.7	50.9	80.5	54.9
OpenL3 (Cramer et al., 2019)	94.9	44.7	75.1	60.4
Wav2CLIP (Wu et al., 2022)	92.9	36.2	75.9	77.0
PANNs CNN14 (Kong et al., 2020)	79.8	N/A	90.9	44.6
PaSST base (Koutini et al., 2022)	94.0	64.0	94.7	N/A
MSM-MAE-200 (16 × 4) (ours)	95.2	50.9	84.3	60.4
MSM-MAE-512 (ours)	96.4	52.2	85.6	69.4

4.3. Experimental Results: Comparison with the HEAR 2021 Results

We compare the results of our two best performing models with the results from the HEAR 2021 Challenge in Tables 2, 3 and 4. We used the HEAR 2021 Challenge results for which

3. We excluded other tasks, such as MAESTRO, because we could not obtain results using the hear-eval-kit.

Table 3: Speech task results.

Model	Lingua10	VoImit	CRM-D	SPC	LbCount
CREPE (Kim et al., 2018b)	14.2	5.1	38.3	21.1	49.9
Wav2CLIP (Wu et al., 2022)	19.2	8.3	51.2	34.7	52.8
KW-MLP (Morshed and Ahsan, 2021)	18.1	5.6	42.4	97.8	45.1
PANNS CNN14 (Kong et al., 2020)	24.4	12.7	55.5	61.8	65.2
PaSST base (Koutini et al., 2022)	25.9	18.2	61.0	63.9	66.0
OpenL3 (Cramer et al., 2019)	33.1	7.8	55.0	76.3	64.1
wav2vec2 (Baevski et al., 2020)	49.3	8.0	65.6	87.9	69.2
SERAB BYOL-S (Scheidwasser-Clow et al., 2022)	45.8	16.0	65.7	94.8	78.5
MSM-MAE-200 (16×4) (ours)	52.9	12.8	73.3	87.3	85.8
MSM-MAE-512 (ours)	50.0	18.3	73.4	86.4	77.8

Table 4: Music task results.

Model	GTZ-M/S	GTZAN	NSPitch	Mrd-Ton	Mrd-Stk	Beijing
PANNS CNN14 (Kong et al., 2020)	99.2	86.0	30.1	82.4	93.9	91.1
Wav2CLIP (Wu et al., 2022)	94.6	74.8	43.9	82.9	94.7	93.6
KW-MLP (Morshed and Ahsan, 2021)	88.9	55.4	60.5	94.2	96.9	91.1
wav2vec2 (Baevski et al., 2020)	94.6	78.0	65.3	82.8	94.3	90.7
CREPE (Kim et al., 2018b)	92.9	64.5	90.0	82.4	89.8	92.8
PaSST base (Koutini et al., 2022)	97.7	88.3	54.1	81.9	96.5	96.6
SERAB BYOL-S (Scheidwasser-Clow et al., 2022)	93.8	83.7	71.2	92.8	97.3	95.3
OpenL3 (Cramer et al., 2019)	96.9	87.9	73.1	93.7	96.7	97.5
MSM-MAE-200 (16×4) (ours)	97.7	86.3	84.0	98.6	97.5	95.3
MSM-MAE-512 (ours)	99.2	86.1	81.2	98.3	97.5	94.9

a single model is used—an ensemble of models is out of the scope of this study—and whose details are available in their papers.

These tables show that our models outperform others in seven tasks: Gunshot Triangulation, Vox Lingua Top 10, Vocal Imitation, CREMA-D, LibriCount, and Mrdingham Tonic and Stroke. Conversely, models specialized in the task outperform our models on other tasks. On environmental sound tasks, FSD50K and ESC-50, the PANNS and PaSST-base outperform ours. These models are supervised learning pre-trained on the AudioSet using its labels. KW-MLP, wav2vec2, and SERAB-BYOLS outperform ours on the Speech Commands task. These models are pre-trained on a speech corpus for specialization in speech. CREPE, specializing in pitch estimation, outperforms ours on the NSynth Pitch task. Except for these specialized models and OpenL3 on the Beijing task, our MSM-MAE models show the top results in most tasks. These results show that MSM-MAE learns audio representation effectively for general tasks without specializing in domains.

4.4. Experimental Results: Impact of Design Choices on Performance

We explore the impact of the design choices of input audio duration and patch size, for which we found positive effects in preliminary experiments. In addition, we also evaluate the performance difference between designs for splitting the input.

4.4.1. IMPACT OF INPUT AUDIO DURATION

Table 5 shows the results of MSM-MAE with various input audio durations. The number at the tail of the model name shows the duration, which is $10 \times$ in ms in actual time (e.g., 96 and 512 are 960 ms and 5.12 seconds, respectively).

These results show that the longer durations yield better results among HEAR 2021 tasks. We see an exceptional behavior with speech tasks (Lingua10, SPC, and LbCount), Gunshot, and GTZ-M/S. However, we think that a long input duration is more beneficial in general-purpose use, because most tasks show the best result with the longer input duration of 400 or 512.

Table 5: Task results for various input audio durations.

(a) Environmental sound tasks						
Model	Gunshot	FSD50K	ESC-50	Beehive		
MSM-MAE-96	98.8	48.7	79.9	54.8		
MSM-MAE-208	89.9	50.8	84.9	52.0		
MSM-MAE-304	90.5	51.7	85.3	65.3		
MSM-MAE-400	90.5	51.8	85.6	62.6		
MSM-MAE-512	96.4	52.2	85.6	69.4		
(b) Speech tasks						
Model	Lingua10	VoImit	CRM-D	SPC	LbCount	
MSM-MAE-96	39.0	12.6	68.0	86.6	78.7	
MSM-MAE-208	48.2	15.7	70.4	87.5	80.3	
MSM-MAE-304	50.6	17.2	72.0	87.3	80.6	
MSM-MAE-400	45.9	16.9	72.8	85.8	80.1	
MSM-MAE-512	50.0	18.3	73.4	86.4	77.8	
(c) Music tasks						
Model	GTZ-M/S	GTZAN	NSPitch	Mrd-Ton	Mrd-Stk	Beijing
MSM-MAE-96	96.9	84.4	81.0	98.2	97.3	89.8
MSM-MAE-208	100.0	84.9	81.3	98.3	97.5	92.8
MSM-MAE-304	98.4	85.6	81.2	98.1	97.5	94.9
MSM-MAE-400	98.5	86.0	81.6	98.5	97.7	94.5
MSM-MAE-512	99.2	86.1	81.2	98.3	97.5	94.9

4.4.2. IMPACT OF PATCH SIZE

Table 6 shows the results of MSM-MAE with various patch sizes for the models that accept 2-s input audio durations: MSM-MAE-208 (16×16), MSM-MAE-200 (16×8), MSM-MAE-200 (16×4), and MSM-MAE-208 (8×16). The 16×16 , 16×8 , 16×4 , and 8×16 show the patch size of F by T .

The results show that the finer time resolutions (16×8 and 16×4) improve performance on 11 out of 15 tasks (Gunshot, FSD50K, Beehive, Lingua10, CRM-D, LbCount, GTZAN, NSPitch, Mrd-Ton, Mrd-Stk, and Beijing). In contrast, the finer frequency resolution (8×16) improves eight tasks (Gunshot, FSD50K, Beehive, CRM-D, LbCount, GTZAN, NSPitch, and Mrd-Ton), and the degree of improvements are smaller than for the finer time resolutions.

Table 6: Task results for various patch sizes. N is the total number of patches.

(a) Environmental sound tasks						
Model	Gunshot	FSD50K	ESC-50	Beehive		
MSM-MAE-208 ($16 \times 16, N = 65$)	89.9	50.8	84.9	52.0		
MSM-MAE-200 ($16 \times 8, N = 125$)	94.0	51.1	83.9	62.0		
MSM-MAE-200 ($16 \times 4, N = 250$)	95.2	50.9	84.3	60.4		
MSM-MAE-208 ($8 \times 16, N = 130$)	96.4	51.2	84.6	62.0		
(b) Speech tasks						
Model	Lingua10	VoImit	CRM-D	SPC	LbCount	
MSM-MAE-208 ($16 \times 16, N = 65$)	48.2	15.7	70.4	87.5	80.3	
MSM-MAE-200 ($16 \times 8, N = 125$)	52.4	14.7	73.1	87.7	84.5	
MSM-MAE-200 ($16 \times 4, N = 250$)	52.9	12.8	73.3	87.3	85.8	
MSM-MAE-208 ($8 \times 16, N = 130$)	48.4	14.1	71.4	87.0	82.8	
(c) Music tasks						
Model	GTZ-M/S	GTZAN	NSPitch	Mrd-Ton	Mrd-Stk	Beijing
MSM-MAE-208 ($16 \times 16, N = 65$)	100.0	84.9	81.3	98.3	97.5	92.8
MSM-MAE-200 ($16 \times 8, N = 125$)	99.2	85.7	81.7	98.6	97.6	93.6
MSM-MAE-200 ($16 \times 4, N = 250$)	97.7	86.3	84.0	98.6	97.5	95.3
MSM-MAE-208 ($8 \times 16, N = 130$)	98.5	86.4	83.2	98.4	97.5	92.8

We can also find the different trends in improvements among models. If we focus on the best results (bold numbers), the finer time resolutions (16×8 and 16×4) excel on speech and music tasks, whereas the finer frequency resolution (8×16) performs better on environmental sound tasks.

In summary, we can improve task performance with finer resolutions using smaller patch sizes, especially with finer time resolutions. However, 16×4 shows that it is not always the case compared to 16×8 ; the finer 16×4 does not clearly show performance improvements superior to that for 16×8 , even though the 16×4 costs quadratically more computational complexity than 16×8 .

4.4.3. IMPACT OF INPUT SPLITTING: PATCHES VS. STRIPS

Table 7 shows the results of MSM-MAE for different designs of splitting the input spectrogram into patches or strips. MSM-MAE-304 (16×16) cuts the input spectrogram into patches along both frequency and time. In contrast, MSM-MAE-304 (80×4) cuts the input into strips along time, simulating the previous methods such as Mockingjay (Liu et al., 2020) that handle the input as a sequence of spectrogram strips. We compare the performance difference between these two ways.

The results show that the patch input model (16×16) outperforms strip input model (80×4) on most tasks. It could indicate that the MIM framework, which takes patches as input and learns both frequency and time structure, is more suitable for learning general-purpose audio representations than the training framework that takes sequential strips as input. By contrast, the strip input model outperforms the patch input model on CREMA-D and SPC tasks, which could also indicate that the strip input is effective for speech tasks, as used in the previous speech self-supervised learning studies that take sequential spectrogram strips as input.

Table 7: Task results for comparing patches vs. strips.

(a) Environmental sound tasks						
Model	Gunshot	FSD50K	ESC-50	Beehive		
MSM-MAE-304 (16 × 16, Patches)	90.5	51.7	85.3	65.3		
MSM-MAE-304 (80 × 4, Strips)	85.7	49.1	81.5	66.0		
difference	4.8	2.6	3.8	-0.7		
(b) Speech tasks						
Model	Lingua10	VoImit	CRM-D	SPC	LbCount	
MSM-MAE-304 (16 × 16, Patches)	50.6	17.2	72.0	87.3	80.6	
MSM-MAE-304 (80 × 4, Strips)	50.0	16.8	73.8	88.1	79.7	
difference	0.6	0.3	-1.9	-0.8	0.9	
(c) Music tasks						
Model	GTZ-M/S	GTZAN	NSPitch	Mrd-Ton	Mrd-Stk	Beijing
MSM-MAE-304 (16 × 16, Patches)	98.4	85.6	81.2	98.1	97.5	94.9
MSM-MAE-304 (80 × 4, Strips)	96.9	85.5	81.0	98.1	97.6	94.5
difference	1.5	0.1	0.1	0.1	-0.1	0.4

4.5. Qualitative Analysis with Visualizations

This section presents analyses based on visualizations to gain an understanding of the representations learned by MSM-MAE. Sections 4.5.1, 4.5.2 and 4.5.3 present visualization of reconstruction results, and Section 4.5.4 shows visualizations of attention maps. We present more visualizations in Appendix A.

4.5.1. RECONSTRUCTIONS WITH RANDOM MASKS

The reconstructions of three sounds in Figure 2 show results similar to those in the MAE paper (He et al., 2022), which reconstruct inputs well but with blurry details, indicating that our models were successfully trained in the experiments. In Figure 2, we observe that the frequency structures, especially the harmonic structures, are reconstructed, and the stationary sounds are easy to reconstruct compared to the short sound events.

Frequency structure reconstructions can be observed by focusing on reconstruction along the frequency axis (vertical axis). Frequency bins are reconstructed even where a few visible patches (white squares) are available in a time frame. Furthermore, we can find clear patterns of the harmonic structures in the reconstruction of sound 1 with trumpet notes. These observations show that the latent representation is encoded effectively to reconstruct frequency bins in a time frame using limited information of a few patches.

We find that stationary sound can be easily reconstructed from sound 2, which is the sound of roaring low-pitch wind. The error of sound 2 shows a lighter color compared to the sound 1 or 3, indicating that the reconstructing of the 2 made a smaller error. In addition, frequency structures are recovered entirely, even without an available visible patch in a time frame, contrary to the sound 1 and 3, where some sound events are not recovered. These observations suggest that the learned representations carry the information related to temporal structure for each sound, such as stationary or short events.

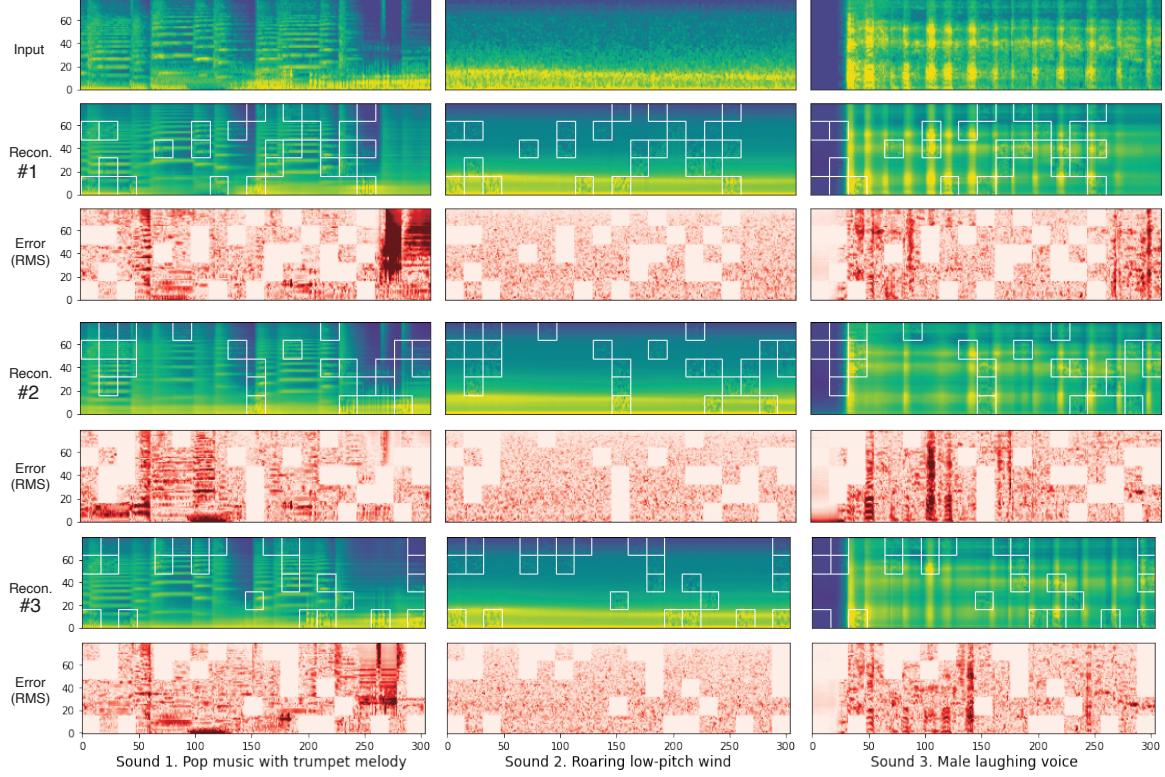


Figure 2: Reconstructions for the pre-trained MSM-MAE-304 with a mask ratio of 0.75, obtained from three attempts of reconstruction of three sounds. Each example shows the input, the reconstruction result, and the difference between them, the error (RMS); the darker the color, the higher the reconstruction error. The white squares in the reconstruction results show the visible patch.

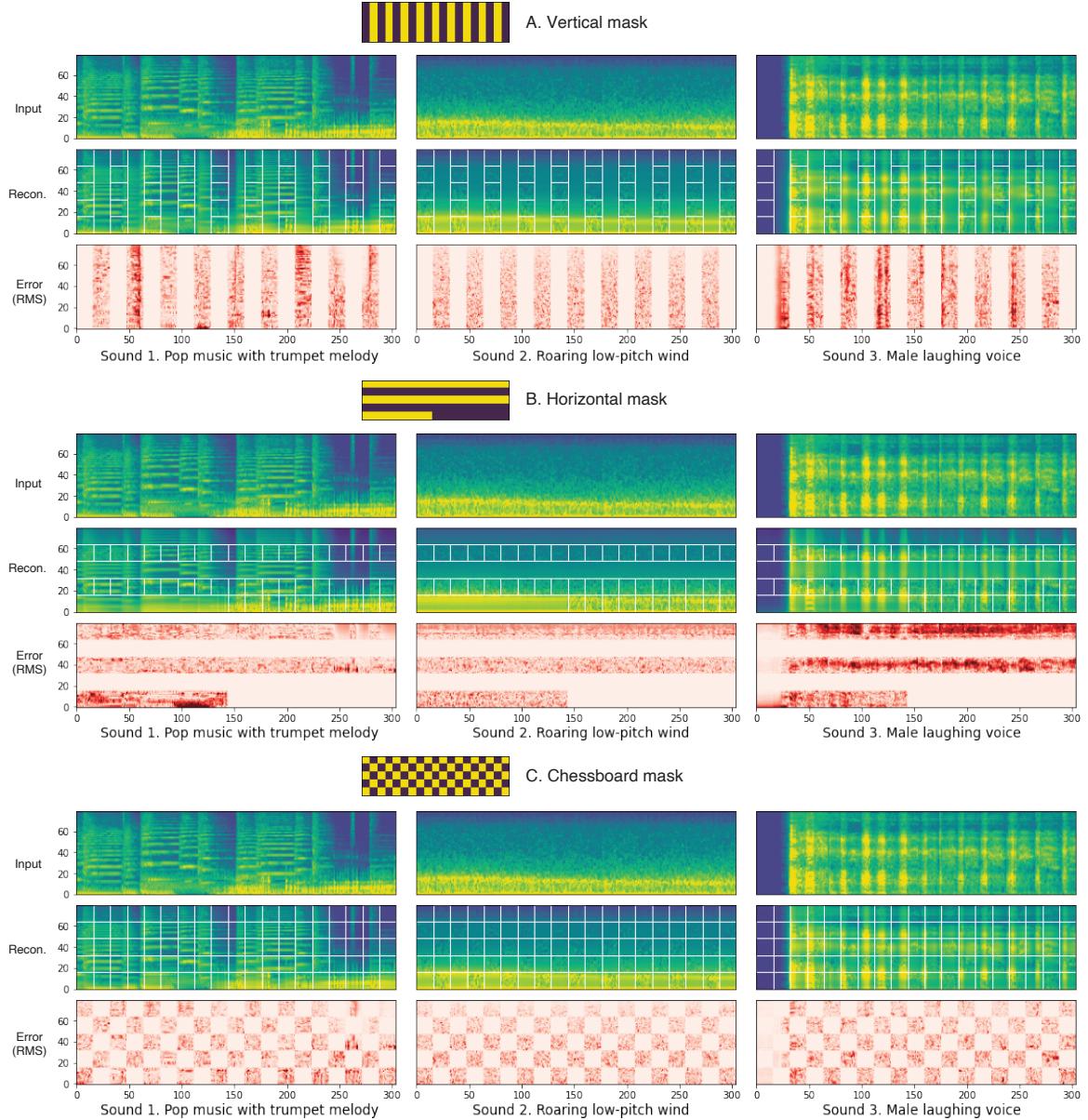


Figure 3: Reconstructions with three patterns of masks, showing the difference in the error under different availability of visible information along axes. The mask ratio is 0.5, i.e., half of the patches are masked for all examples.

4.5.2. RECONSTRUCTIONS WITH PATTERNED MASKS

We compare the difference in reconstruction under the different availability of visible information made by the three mask patterns in Figure 3. In A, vertical mask, visible and mask patches alternate along the time axis; models are to recover masked patches using visible patches adjoining on the time axis. In B, horizontal mask, they alternate along the frequency axis; visible patches adjoining in the frequency axis are available. In C, the chessboard mask is for making visible patches available around the masked patches. All cases have the same mask ratio of 0.5.

In Figure 3, we observe that the more adjoining visible patches available, the easier the reconstruction becomes. For all the sounds, the errors show lighter color on the chessboard mask results than that on the other vertical and horizontal mask results, showing the smaller reconstruction error for the chessboard mask, where more adjoining visible patches are available.

If we focus on the horizontal mask in B, we can observe that harmonic structures are reconstructed more easily than noises. Sound 1, a rich harmonic structure of trumpet notes, shows less reconstruction error than sound 3, with less structured frequency patterns of noises in laughing voices. This observation suggests that the learned representation encodes the information of the harmonic structure more effectively than that of the noise.

4.5.3. RECONSTRUCTIONS WITH VARIOUS MASK RATIOS

We varied the mask ratio to observe how the reconstruction changes according to it. We used the 3-s model (MSM-MAE-304) to reconstruct the input spectrograms with varied mask ratios from 0.40 to 0.99, focusing on cases with extremely small numbers of visible patches (e.g., 1, 2, 5, and 10 visible patches, corresponding to mask ratios of 0.99, 0.98, 0.95, and 0.90, respectively).

Figure 4 shows the example reconstruction results. The results show that reconstruction succeeds entirely up to the default ratio of 0.75, whereas it degrades noticeably at higher ratios. As the mask ratio increases, only the patches around the visible patch are reconstructed relatively accurately, while the rest become blurry copies of the pattern around the nearest visible patch.

Focusing on the mask ratio of 0.99, which encodes only one visible patch, we can observe the reconstruction of both local and global patterns. Locally, the frequency pattern of the time frames around the visible patch is reconstructed relatively clearly. Globally, a frequency pattern similar to the average of the original spectrogram is reconstructed stationary over the entire spectrogram. This observation suggests that even though only one patch was encoded, information related to both local and global patterns learned from the training dataset is embedded in the representation.

4.5.4. SELF-ATTENTION MAP VISUALIZATIONS

We visualize the self-attention of the encoder of pre-trained MSM-MAE-304 for six sounds in Figures 5 and 6. We picked two reference points each to show self-attention maps. These self-attention maps average from all heads in the last layer.

We see that the self-attention map reflects the repetition or continuation in the input sounds. Sound 1 of pop music has a clear repetition of notes in the input, which we can

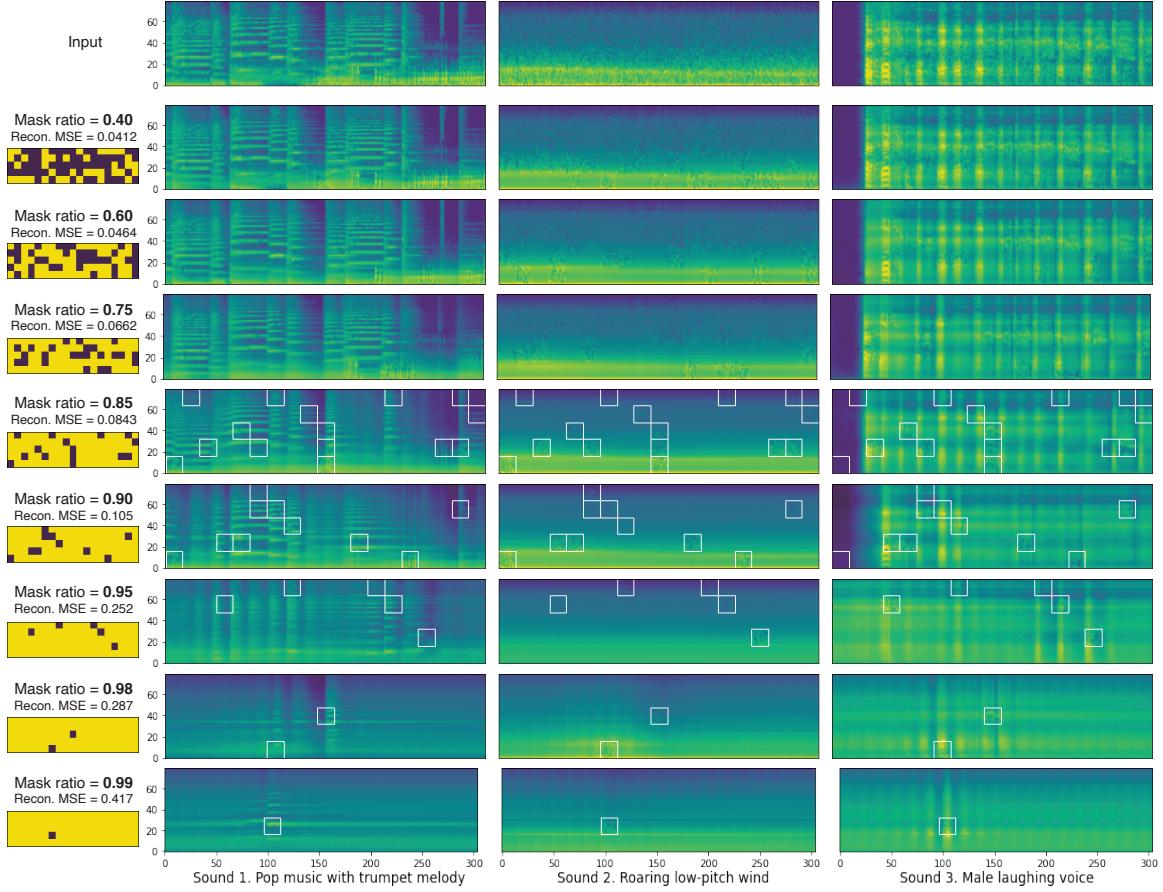


Figure 4: Reconstructions with various mask ratios, focusing on cases of extremely small numbers of visible patches. We show white squares, the visible patches, on the results only with mask ratios higher than 0.75 to improve visibility that avoid the figure becoming too crowded with white boxes to tell which was reconstructed or not. MSE is the mean squared error of reconstruction averaged over the entire spectrogram.

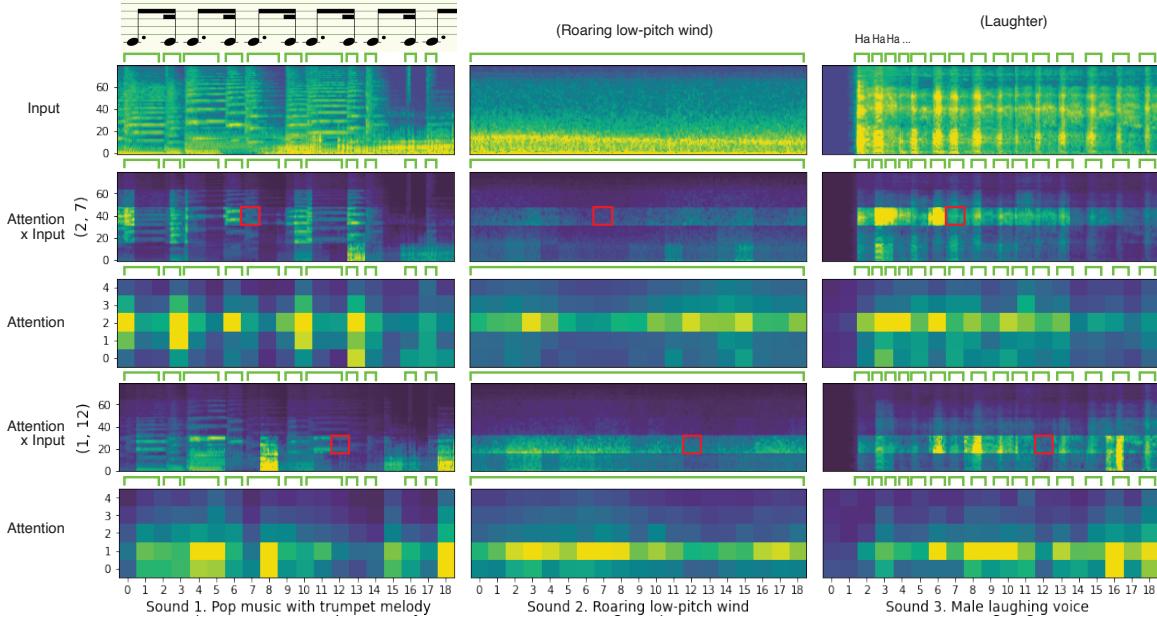


Figure 5: MSM-MAE encoder self-attention map for reference points (2, 7) and (1, 12) depicted as red boxes, showing the strength of attention from a token of interest (reference point) to all other tokens.

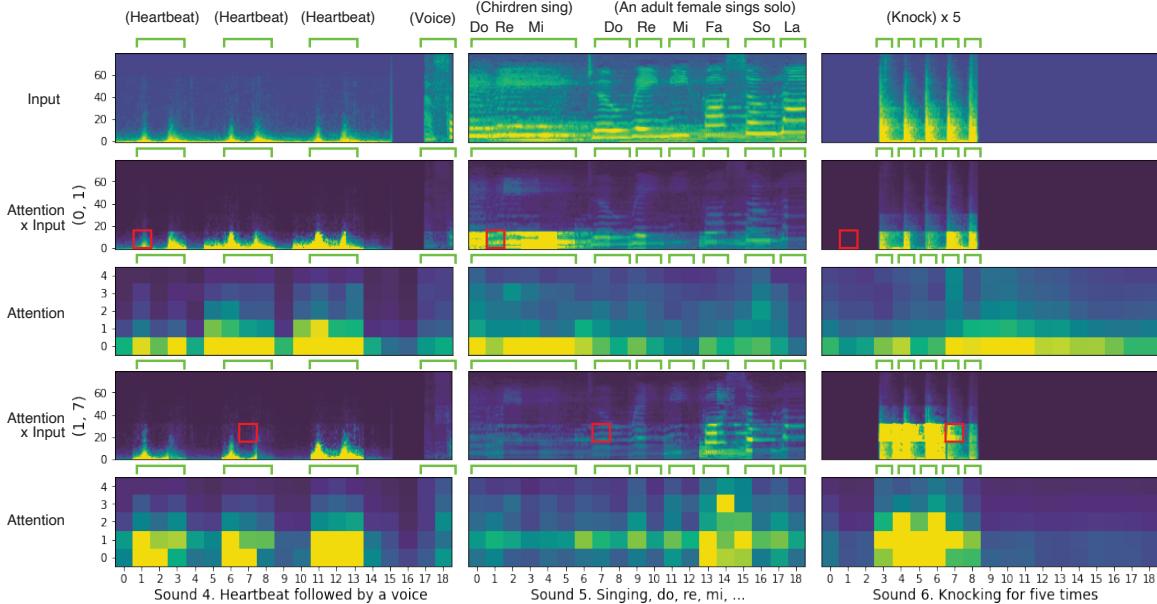


Figure 6: MSM-MAE encoder self-attention map for reference points (0, 1) and (1, 7) depicted as red boxes.

also find in the self-attention map. On the other hand, sound 2 is a stationary sound of blowing wind, and the attention continues along time similarly. Interestingly, we find in the first reference point in sound 1 that the self-attentions are strong at the same position in the beat, even though there are two similar notes per beat.

We also observe coarse segmentation of similar sounds for the reference points. For example, sound 4 shows that heartbeats are segmented in the self-attention maps of both reference points, less attending to the following voice; sound 5 shows that the first half of children’s singing voices are segmented in the first reference point, whereas the latter half of an adult female voice is coarsely segmented in the second reference point.

5. Conclusion

In this paper, we sought to learn audio representation from the input itself as supervision by using a pretext task of modeling masked spectrogram patches, which we call Masked Spectrogram Modeling (MSM). To implement MSM, we employed Masked Autoencoders (MAE) with audio spectrogram as input.

We conducted evaluations on the HEAR 2021 NeurIPS Challenge using its benchmark suite across a variety of domains, including speech, environmental sound, and music. Our models outperformed the HEAR 2021 Challenge results on seven out of 15 tasks (e.g., accuracies of 73.4% on CREMA-D and 85.8% on LibriCount) while showing top performances on other tasks where specialized models perform better. In addition, we investigated design choices of input audio duration and patch size and confirmed that longer duration and finer time resolution with a smaller patch size improve performance.

We also conducted qualitative analyses on various visualizations of outputs from both the MAE encoder and decoder. We observed frequent and temporal structures in the reconstruction results and the self-attention maps, suggesting that the learned representations hold information related to these structures.

While this study does not provide an exhaustive exploration, the quantitative results proved the effectiveness of MSM using MAE, and the qualitative observations suggested useful information in the learned representations. We believe that the MSM framework holds promising further improvements and applications in the future. Future directions may include exploring ViT model settings, which we fixed in this study, suitable for an audio spectrogram, and exploring other MAE variants and extensions potentially ideal for learning audio representations.

References

- Akshay Anantapadmanabhan, Ashwin Bellur, and Hema A Murthy. Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization. In *ICASSP*, pages 181–185, 2013.
- Alan Baade, Puyuan Peng, and David Harwath. MAE-AST: Masked Autoencoding Audio Spectrogram Transformer. In *Interspeech*, pages 2438–2442, 2022.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*, 2020.

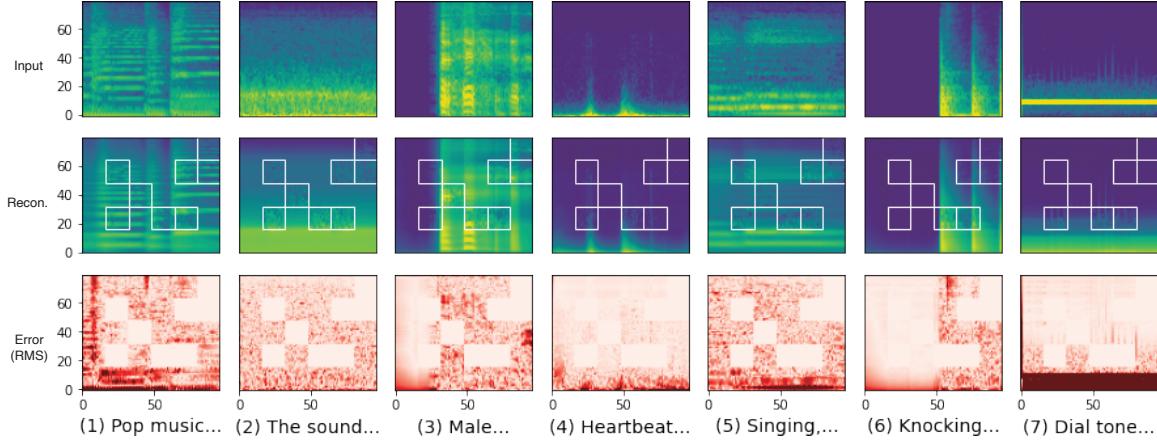
- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.
- Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE Trans. Affective Comput.*, pages 377–390, 2014.
- Shuhao Cao, Peng Xu, and David A. Clifton. How to understand masked autoencoders. *arXiv preprint arXiv:2202.03670*, 2022.
- Po-Han Chi, Pei-Hung Chung, Tsung-Han Wu, Chun-Cheng Hsieh, Yen-Hao Chen, Shang-Wen Li, and Hung-yi Lee. Audio albert: A lite bert for self-supervised learning of audio representation. In *SLT*, pages 344–350, 2021.
- Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng. Masked spectrogram prediction for self-supervised audio pre-training. *arXiv preprint arXiv:2204.12768*, 2022.
- Seth Cooper and Steven Shaw. Gunshots recorded in an open field using ipod touch devices, 2020. URL <https://doi.org/10.5061/dryad.wm37pvmkc>.
- J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello. Look, listen and learn more: Design choices for deep audio embeddings. In *ICASSP*, pages 3852–3856, 2019.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with WaveNet autoencoders. In *ICML*, pages 1068–1077, 2017.
- Eduardo Fonseca, Diego Ortego, Kevin McGuinness, Noel E. O’Connor, and Xavier Serra. Unsupervised contrastive learning of sound event representations. In *ICASSP*, pages 371–375, 2021.
- Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: An open dataset of human-labeled sound events. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 30:829–852, 2022.
- J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, pages 776–780, 2017.
- Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *AAAI*, volume 36, pages 10699–10709, 2022.

- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988, 2022.
- S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson. Cnn architectures for largescale audio classification. In *ICASSP*, pages 131–135, 2017.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech, Language Process.*, page 3451–3460, 2021.
- Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *arXiv preprint arXiv:2207.06405*, 2022.
- Bong Rok Kim, Madhav Ghei, Bryan Pardo, and Zhiyao Duan. Vocal imitation set: a dataset of vocally imitated sound events using the audioset ontology. In *DCASE*, 2018a.
- Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *ICASSP*, pages 161–165, 2018b.
- 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 Trans. Audio, Speech, Language Process.*, 28:2880–2894, 2020.
- Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. In *Interspeech*, pages 2753–2757, 2022.
- Andy T. Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP*, pages 6419–6423, 2020.
- Andy T. Liu, Shang-Wen Li, and Hung-yi Lee. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Trans. Audio, Speech, Language Process.*, page 2351–2366, 2021.
- Mashrur M. Morshed and Ahmad Omar Ahsan. Attention-free keyword spotting. *arXiv preprint arXiv:2110.07749*, 2021.
- Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *IJCNN*, pages 1–8, 2021.
- Ines Nolasco, Alessandro Terenzi, Stefania Cecchi, Simone Orcioni, Helen L. Bear, and Emmanouil Benetos. Audio-based identification of beehive states. In *ICASSP*, pages 8256–8260, 2019.
- Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *ACM-MM*, pages 1015–1018, 2015.

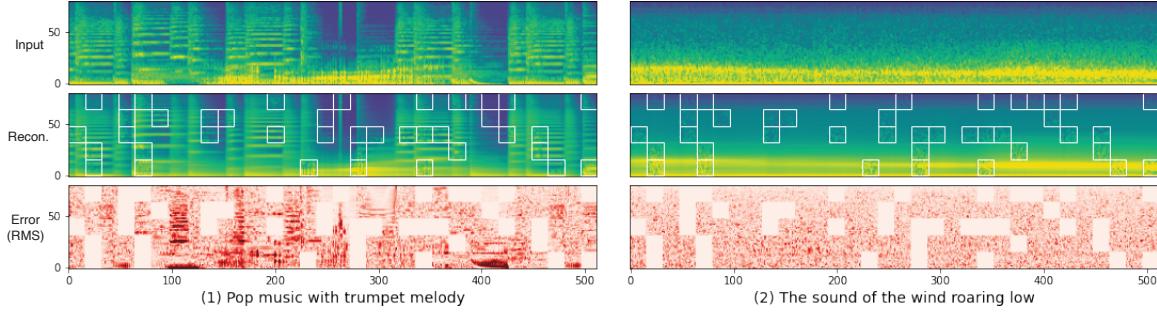
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021.
- Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP*, pages 3875–3879, 2021.
- Neil Scheidwasser-Clow, Mikolaj Kegler, Pierre Beckmann, and Milos Cernak. Serab: A multi-lingual benchmark for speech emotion recognition. In *ICASSP*, pages 7697–7701, 2022.
- Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Félix de Chaumont Quirky, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, and Yinnon Haviv. Towards learning a universal non-semantic representation of speech. In *Interspeech*, pages 140–144, 2020.
- Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. In *ISMIR*, pages 673–681, 2021.
- Fabian-Robert Stöter, Soumitro Chakrabarty, Bernd Edler, and Emanuël A. P. Habets. Countnet: Estimating the number of concurrent speakers using supervised learning. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 27(2):268–282, 2019.
- Mi Tian, Ajay Srinivasamurthy, Mark Sandler, and Xavier Serra. A study of instrument-wise onset detection in beijing opera percussion ensembles. In *ICASSP*, pages 2159–2163, 2014.
- Joseph Turian, Jordie Shier, Humair Raj Khan, Bhiksha Raj, Björn W. Schuller, Christian J. Steinmetz, Colin Malloy, George Tzanetakis, Gissel Velarde, Kirk McNally, Max Henry, Nicolas Pinto, Camille Noufi, Christian Clough, Dorien Herremans, Eduardo Fonseca, Jesse Engel, Justin Salamon, Philippe Esling, Pranay Manocha, Shinji Watanabe, Zeyu Jin, and Yonatan Bisk. HEAR: Holistic Evaluation of Audio Representations. In Douwe Kiela, Marco Ciccone, and Barbara Caputo, editors, *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, volume 176 of *Proceedings of Machine Learning Research*, pages 125–145. PMLR, 06–14 Dec 2022. URL <https://proceedings.mlr.press/v176/turian22a.html>.
- G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Speech Audio Process.*, 10(5), 2002.
- Jörgen Valk and Tanel Alumäe. Voxlingua107: A dataset for spoken language recognition. In *SLT*, pages 652–658, 2021.
- Luyu Wang, Pauline Luc, Yan Wu, Adrià Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, and Aäron van den Oord. Towards learning universal audio representations. In *ICASSP*, pages 4593–4597, 2022.
- P. Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *arXiv preprint arXiv:1804.03209*, April 2018.
- Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP*, pages 4563–4567, 2022.

Appendix A. Reconstruction Examples of Various MSM-MAE Models

Figures 7 and 8 visualize the reconstruction examples of various MSM-MAE models.



(a) Examples of the MSM-MAE-96 (input duration: 960 ms).



(b) Examples of the MSM-MAE-512 (input duration: 5.12 seconds).

Figure 7: Reconstructions of the models for short and long input audio duration.

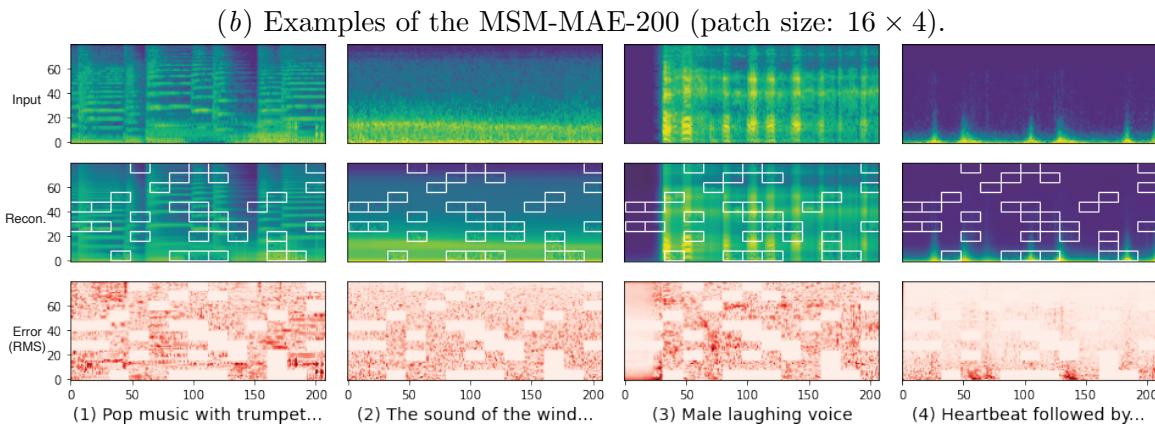
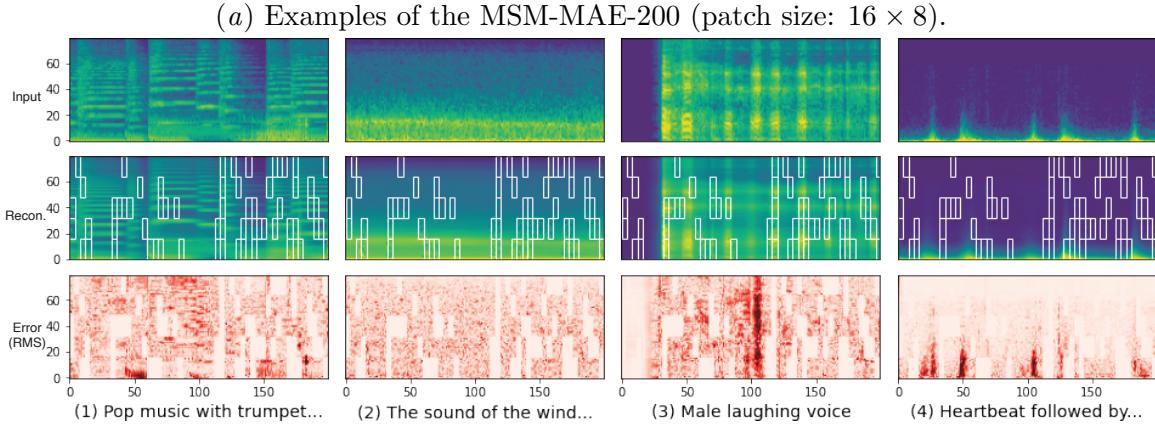
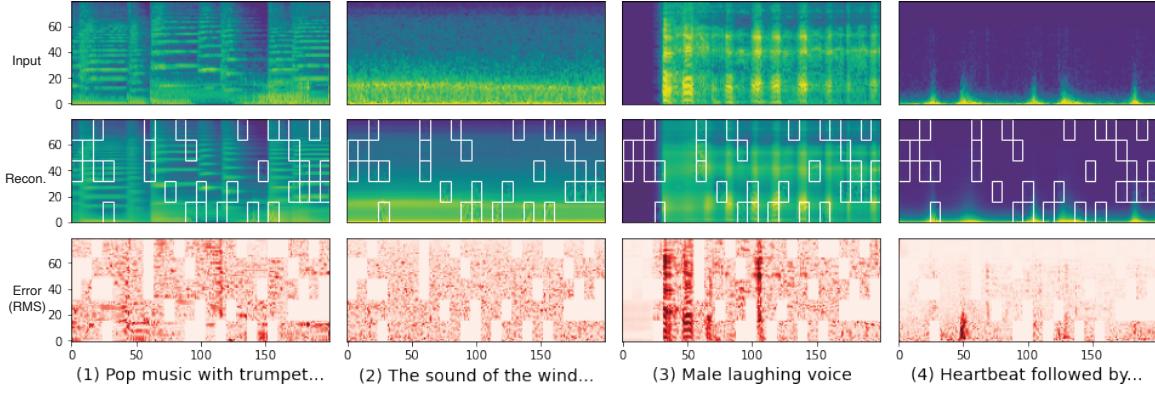


Figure 8: Reconstruction examples of the models with various patch sizes.

Appendix B. Reconstruction Examples of Vocal Imitation Task

We visualize the reconstruction examples of the Vocal Imitations (Kim et al., 2018a) task, on which experimental results show much lower performance than on other tasks.

As shown in Figure 9, we can confirm that the Vocal Imitation samples can be reconstructed similarly well to the AudioSet samples in Figure 2. Nevertheless, Table 3 shows that Vocal Imitation results are very low compared to the other tasks for all models, suggesting that the poor performance is due to task-specific difficulty.

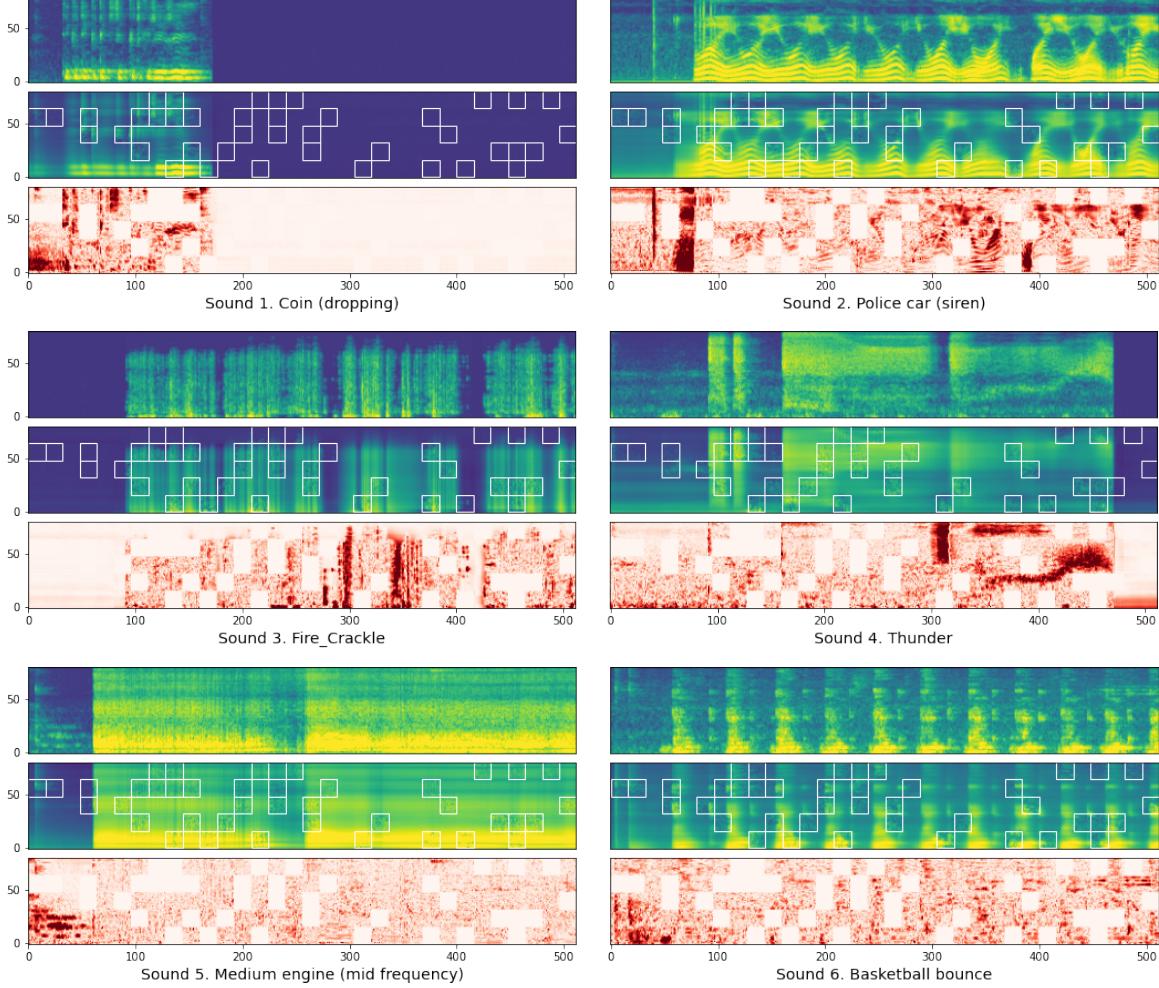


Figure 9: Reconstructions of the Vocal Imitations samples with the MSM-MAE-512.