Aalborg Universitet

Masked Autoencoders with Multi-Window Local-Global Attention Are Better Audio Learners

Yadav, Sarthak; Theodoridis, Sergios; Hansen, Lars Kai; Tan, Zheng-Hua

Masked Autoencoders with Multi-Window Local-Global Attention Are Better Audio Learners

Sarthak Yadav[1,2]   **Sergios Theodoridis**[1,4]   **Lars Kai Hansen**[2,3]   **Zheng-Hua Tan**[1,2]

[1] Department of Electronic Systems, Aalborg University, Denmark
[2] Pioneer Centre for Artificial Intelligence, Denmark
[3] Department of Applied Mathematics and Computer Science, DTU, Denmark
[4] Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece
`[sarthaky,sthe,zt]@es.aau.dk, lkai@dtu.dk`

Abstract

In this work, we propose a Multi-Window Masked Autoencoder (MW-MAE) fitted with a novel Multi-Window Multi-Head Attention (MW-MHA) module that facilitates the modelling of local-global interactions in every decoder transformer block through attention heads of several distinct local and global windows. Empirical results on ten downstream audio tasks show that MW-MAEs consistently outperform standard MAEs in overall performance and learn better general-purpose audio representations, along with demonstrating considerably better scaling characteristics. Investigating attention distances and entropies reveals that MW-MAE encoders learn heads with broader local and global attention. Analyzing attention head feature representations through Projection Weighted Canonical Correlation Analysis (PWCCA) shows that attention heads with the same window sizes across the decoder layers of the MW-MAE learn correlated feature representations which enables each block to independently capture local and global information, leading to a decoupled decoder feature hierarchy. Code for feature extraction and downstream experiments along with pre-trained models will be released publically.

1   Introduction

With rapid advances in hardware capabilities driving models of ever-increasing complexity and appetite for data, leveraging unlabelled data for learning effective deep representations has garnered significant interest. Self-supervised learning, which solves a pretext task that utilizes labels generated from the data itself, has emerged as a notable approach for training deep neural representations without labelled data. Several methods for learning self-supervised representations from audio data have been proposed, including autoregressive methods that learn to predict the future based on the past input [1, 2, 3, 4, 5], methods that learn contrastive representations from different views of the input [6, 7, 8, 9, 10, 11, 12], and masked predictive modelling methods that learn to predict removed portions of the input data [13, 14, 15].

Together with the transformer architecture [16] and its successors [17, 18], *masked predictive modelling* has led to significant advances across natural language processing (NLP) [13, 19], computer vision [15, 20] and audio and speech processing [14, 21, 22]. Masked Autoencoders (MAEs) by [23] are a recent addition to the masked predictive modelling family. Initially proposed for learning visual representations from randomly masked image patches, MAEs are experiencing widespread adoption across several domains [24, 25, 26, 27, 28, 29] due to their inherent scalability and simple
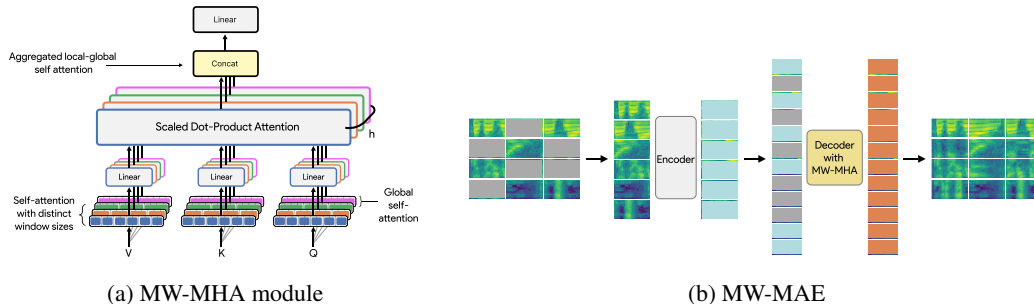
| (a) MW-MHA module | (b) MW-MAE |

Figure 1: An overview of the proposed Multi-Window Multi-Head Attention (MW-MHA) module, and the overall MW-MAE architecture. In MW-MHA, each attention head operates on non-overlapping windows of different sizes (coded by different colours) of the input matrices. As evident from b) MW-MAE uses the proposed MW-MHA block only in the decoder.

design. In the audio domain, several recent works have adapted MAEs to learn a general-purpose audio representation from spectrogram inputs [30, 31]. These works address several challenges that are unique to the audio domain and exhaustively study the effect of masking strategies and other hyperparameters, providing vital information for training MAEs on audio data.

Recent works have shown that leveraging local information in the Multi-Head Attention (MHA) module of a transformer layer through convolutions [32], attention with local windows [18, 33] or pooling attention [34, 35, 36] can lead to improved performance. Within the framework of Masked Autoencoders, [37] evaluated the impact of local windowed attention [18] for audio representation learning, demonstrating better performance across 4 downstream audio recognition tasks. However, in these methods: (i) all the attention heads within a MHA module operate at the same local context, thus only capturing local information at the transformer layer level, and (ii) they require explicit approaches to mitigate the lack of connections across windows and to capture local-global information.

In this work, we propose Multi-Window Masked Autoencoders (MW-MAE) with both local and global attention for learning general-purpose audio representations from spectrogram inputs. Decoders in an MW-MAE are fitted with a novel Multi-Window Multi-Head Attention module (MW-MHA) (Fig 1). Each attention head in the proposed MW-MHA module computes self-attention over non-overlapping windows of different sizes, which facilitates modelling of local-global interactions in every decoder transformer block. The proposed MW-MAEs outperform standard MAEs on 10 downstream audio recognition tasks. At the same time, MW-MAEs adapt better to varying patch sizes and increasing number of patches, perform better in low-data scenarios, as well as demonstrate better performance and scaling characteristics with respect to changing encoder and decoder complexities. Exhaustive exploratory analysis of attention distances and entropies shows that attention heads in MW-MAE encoders learn broader local-global attention as compared to standard MAEs, despite having an identical architecture. Further, analysis of feature representations from the decoder attention heads using Projection Weighted Canonical Correlation Analysis (PWCCA) [38] indicates that attention heads with the same window sizes across the decoder layers of the MW-MAE learn correlated feature representations leading to a decoupled feature hierarchy, confirming that MW-MHA modules learn local-global features in each decoder block.

2 Background and Related Works

Recently, several works have been proposed for learning audio representations in a self-supervised manner. Most of these works can be loosely categorized into one or more of the following groups based on their underlying pretext task: (i) predictive; (ii) contrastive; and (iii) masked predictive modelling. Several methods adopt a predictive coding approach, which aligns well with the sequential nature of audio input. Autoregressive predictive coding (APC) [2, 4, 3] is one such method which utilizes Recurrent Neural Networks (RNNs) to predict future elements of a sequential input given the past, whereas non-autoregressive approaches using Convolutional Neural Networks (CNNs) have also been proposed [5]. Contrastive predictive coding [1] optimizes a predictive coding objective in the latent space while simultaneously optimizing a contrastive objective function. This brings us

to contrastive representation learning, which operates on the premise of learning a representation space that maximizes agreement between views from the same input sample while minimizing inter sample agreement. Several contrastive learning-based methods, originally proposed for computer vision [39, 40, 41], have been adapted for learning audio representations [7, 42, 43]. A widely used contrastive approach for learning speech representations is the Wav2Vec family of algorithms [9, 44, 10, 11], which learn contextualized speech representations by optimizing a contrastive objective between quantized latent representations and representations generated from masked time steps. Finally, self-supervised learning methods based on masked predictive modelling have a simple premise: remove a portion of the input data, and learn to predict the removed content. After being (re-)popularized by the likes of BERT [13] in NLP and fueled by the recognition of the Transformer [16] as a viable cross-domain neural architecture, masked modelling has seen wide adoption in several domains, such as computer vision [20, 15], audio and speech [45] as well as a multi-domain self-supervised learning frameworks [46]. In the audio domain, several recent methods use masked predictive modelling to learn self-supervised representations [44]. These include HuBERT [14], which trains a BERT-like model to predict pre-determined cluster assignments from masked speech features, WavLM [21], which learns a joint denoising and masked prediction task, and SSaST [22], which jointly reconstructs and contrasts masked patches. More recently, [47] proposed an iterative masked modelling approach using an iterative self-distilled tokenizer that generates refined discrete labels from audio input data for the next stage of pretraining.

Masked Autoencoders (MAEs): Recently, [23] proposed Masked Autoencoders for learning self-supervised image representations. In an MAE, the input is split into non-overlapping patches, which are then linearly projected to a fixed dimension by the patch embedding layer. A large subset of these patches is masked out (*e.g.*, over 75% of the patches), and the unmasked patches are then encoded by a Vision Transformer (ViT) [17]. With learnable mask tokens filled in the correct positions to restore the original patch order, these encoded patches are fed to a transformer based decoder, whose objective is to learn to reconstruct the masked patches. The high masking ratio allows large encoders to be paired with significantly smaller decoders due to the reduced encoding complexity, while simultaneously forcing the encoder to learn better contextualized representations by minimizing extrapolation from redundant neighbouring patches. Several recent works based on MAEs have been proposed for training general-purpose audio representations. [30] explored a joint discriminative and generative objective for training audio MAEs and evaluated fine-tuning performance on 5 downstream tasks. By training shallow downstream classifiers on 15 downstream tasks in accordance with the HEAR-2021 [48] protocol, [31] investigated various hyper-parameters such as patch size and the effect of input audio clip duration on model performance. [37] investigated shifting windows based local self-attention [18] of a fixed context (4×4 windows) in all but the last few layers of audio MAEs. In contrast, in this work we propose a Masked Autoencoder fitted with a novel Multi-Window Multi-Head Attention module that can model attention at several context levels and can capture local-global interactions in every transformer layer.

3 Proposed Approach

3.1 Multi-Window Multi-Head Attention

To better capture local-global attention, we propose a Multi-Window Multi-Head Attention (MW-MHA) module, where each attention head computes self-attention across spectrogram patches in different local windows and then combines the contribution of each attention head, as illustrated in Figure 1a. We define MW-MHA with h parallel heads as follows:

$$\text{MWMHA}(Q, K, V) = \text{Concat}(\text{winHead}_1, ..., \text{winHead}_h)W^O \tag{1}$$

$$\text{winHead}_i = \text{WinAttention}(QW_i^Q, KW_i^K, VW_i^V, win_i) \tag{2}$$

As opposed to MHA, each head winHead_i computes local self-attention over non-overlapping windows of size win_i by partitioning input matrices $QW_i^Q, KW_i^K, VW_i^V \in \mathbb{R}^{n \times d_k}$ into $Q_{win_i}, K_{win_i}, V_{win_i} \in \mathbb{R}^{m \times win_i \times d_k}$, given that $n = m \times win_i$. This is followed by computing standard self-attention $X_{win_i} = \text{Attention}(Q_{win_i}, K_{win_i}, V_{win_i})$ [16] on these partitioned inputs. Finally, $X_{win_i} \in \mathbb{R}^{m \times win_i \times d_k}$ is reshaped to $X \in \mathbb{R}^{n \times d_k}$ to get the output.

In the proposed MW-MHA module, individual attention heads capture information at multiple local contexts, and the final projection matrix $W_i^O \in \mathbb{R}^{hd_k \times d_m}$ learns the contribution of each of these heads, allowing inter-window interaction and connection. This design facilitates learning both *local* and *global* time-frequency information at several granularities in every transformer block (as supported by exploratory analysis in Section 5). This is in contrast to shifting [18, 49], striped [33] windowed self-attention, or pooling attention [34, 35, 36], where all attention heads within the same block have the same window size and thus only perform local self-attention at the block level. Pseudo-code for the proposed MW-MHA is provided in Appendix A.

3.2 Masked Autoencoder with Multi-Window Multi-Head Attention

Patch embeddings, masking strategy and masking ratio: We use mel-spectrograms as inputs, partitioning them into non-overlapping patches, which are then flattened and embedded into linear projections. For encoding positional information, we use fixed sinusuidal positional embeddings, similar to [30, 31, 37]. We use a high masking ratio (80%) and random unstructured masking, which have been shown to work well for audio [31, 37].

Encoder: In line with previous work [23, 31, 37], we use a Vision Transformer (ViT) [17] based encoder, which only processes non-masked patches (20% in this work). Due to the random masking strategy, majority of the patches are not processed by the encoder at training time. This minimizes the benefit of using the proposed MW-MHA modules in the encoder transformer blocks (as evidenced by experiments in Section 4.4). Thus, transformer blocks in our encoder use standard Multi-Head Attention.

Decoder with Multi-Window Multi-Head Attention: We add fixed sinusoidal positional embeddings to the encoded visible patches concatenated with trainable *masked tokens* after restoring original patch order. The resulting tensor is then fed to the decoder, which is also a stack of transformer layers, followed by a linear head that reconstructs the original input spectrogram. This is consistent with previous works [23, 31, 37]. Given that the decoder processes all the patches, we replace the Multi-Head Attention module with the proposed Multi-Window Multi-Head Attention, thus modelling local-global attention in every decoder block.

Selecting window sizes: We follow a simple rule for determining the window sizes of each constituent $\mathrm{winHead}_i$: given the total number of patches n_p, we simply take all non-unary factors of n_p and add two additional global self-attention heads. As an example, our default configuration yields $n_p = 250$, and thus the window sizes for each MW-MHA module in all decoder blocks will be $[2, 5, 10, 25, 50, 125, 250, 250]$ for a total of 8 attention heads, which is a reasonable number of attention heads inline with previous research [23, 37]. Not only is this method simple to follow, but it also scales well with number of patches, effectively covering several possible local context levels.

Pre-training objective: During pre-training, we optimize a loss function that computes mean squared error (MSE) between the predicted masked patches and their corresponding input spectrogram patches. In early experiments, we observed reduced performance when using per-patch normalization, and thus we do not normalize target spectrogram patches.

4 Experiments

4.1 Datasets and Tasks

Pre-training: We use the full AudioSet dataset [50] (AS-5k) for pre-training MAEs and MW-MAEs. With over 5000 hours of audio data distributed in 2 million 10-second weakly annotated YouTube clips spanning 527 classes, AudioSet is one of the largest publicly available audio corpora.

Downstream tasks: Recently, several standardized benchmarks have been proposed to evaluate audio representations thoroughly across a wide variety of domains, such as SUPERB [51] and HEAR [48]. While both benchmarks offer avenues for fast, reproducible and accessible comparison of audio representations, the SUPERB benchmark focuses primarily on speech-processing applications. In contrast, the HEAR benchmark consists of 19 tasks spanning diverse audio domains of speech, music and environmental sounds and redistributes standardized and preprocessed datasets. However, some of these tasks are simply smaller subsets of one another, whereas performance on some HEAR tasks has been demonstrated to be correlated [48]. For evaluating audio representations, we utilize a

subset of the HEAR benchmark which consists of ten diverse tasks spanning multiple domains, which can be found in Appendix B along with the underlying selection criterion. We believe the selected tasks constitute a balanced evaluation protocol that facilitates assessment of audio representations without doing excessive evaluations. For downstream evaluation, we follow the HEAR protocol, where for each task, a shallow downstream classifier is trained on top of fixed features extracted using a pretrained model. This practice has become quite prevalent and allows the evaluation of how representations generalize to a broad range of tasks without the drawbacks of fine-tuning large, heterogeneous neural networks.

Measuring overall performance: Given the wide variety of downstream tasks and feature representations evaluated, a single metric to quantify the performance would significantly aid analysis. However, given the differing difficulty levels of the tasks as well as outliers arising from the nature of the representations evaluated, simply averaging the scores is not sufficient. To counteract this, we utilize a normalized overall score to track overall performance of a given audio representation. Mathematically, overall score $s(m) \in [0., 100.]$ of a model m is given as:

$$s(m) = \frac{1}{|T|} \sum_{t \in T} \frac{x_t(m) - min_t}{max_t - min_t} * 100 \tag{3}$$

where $x_t(m)$ denotes performance of the model m on task t, and min_t and max_t represent the worst and the best performance across all models on the task. By taking the relative performance of the best and the worst approach on a task into consideration, this overall score takes *how hard the task is to improve on* in consideration. It is worth noting that the normalized score is computed across all the evaluated methods in all upcoming sections, including ablations. This is similar to the overall score used by the public leaderboard of the SUPERB [51] benchmark, except that we do not set the normalized value of the worst performing method to 0, and the proposed overall score has an upper range of 100.0.

4.2 Implementation Details

Features: We use log-scaled mel spectrograms with a window size of 25 ms, a hop size of 10 ms and $F = 80$ mel-spaced frequency bins in the $50 - 8000$ Hz range, extracted using the *torchaudio* [52] toolkit. All datasets have a sampling frequency of 16000 Hz. Instead of normalizing by dataset statistics, we adopt a per-instance standardization scheme.

Pre-training: We use the AudioSet dataset for pre-training our Masked Autoencoders. We extract log-scaled mel spectrograms for the entire AudioSet dataset and randomly crop a segment 200 timesteps in length from each data sample. Our default configuration consists of a ViT-B encoder. All our MAE variants accept a 200×80-dimensional (T $\times$ F, respectively) input corresponding to an audio duration of 2 seconds, which achieves performance on-par with longer input durations as demonstrated by [31]. For our default configuration, our patch embedding computes non-overlapping patches with a patch size of (4×16), given it's desirable performance v/s complexity tradeoff as found by [31]. A key characteristic of the Masked Autoencoder paradigm is its asymmetric design, which allows pairing small decoders with large encoders while scaling favourably for linear probe performance [23]. Thus, in contrast to [37], we adopt a smaller 4-layer deep transformer-based decoder of width 384 and 8 attention heads for our default configuration. We train Masked Autoencoders with the proposed MW-MHA module, which are referred to as MW-MAEs, as well as their standard MAE counterparts. All MAEs are pre-trained for 100 epochs with a batch size of 1024 and a weight decay of 0.05 on a single TPU-v3 VM with 8 TPU cores, with the default configuration taking around 36 hours to train. We warm up for ten epochs to a base learning rate of 1e-5, followed by a cosine decay schedule. A masking ratio of 0.8 with unstructured random masking is used, and no other data augmentations are used during pre-training.

Training downstream models: We first extract fixed feature embeddings for all downstream tasks to train downstream models. In the MAE framework, the decoder is discarded after pretraining and feature embeddings are extracted using just the encoder. To generate scene embeddings consistent with the HEAR protocol, we use the exact patch aggregation process as [31]: we break audio clips into non-overlapping 2 second chunks, concatenating the features in time and finally taking a mean over the time axis to generate a fixed vector representation independent of the input audio duration. The *hear-eval-kit*, released alongside the HEAR benchmark, was used to extract fixed feature embeddings

Table 1: Comparison with various audio representations from the literature. 95% confidence intervals are reported over 10 runs on downstream classifiers. We pre-trained all highlighted audio representations, with different gray levels indicating directly comparable MAE and MW-MAE configurations. For other pre-trained audio representations, publicly available official implementations were used. All downstream models were trained by us using the *hear-eval-kit*. $s(m)$ denotes the proposed normalized overall score (Sec 4) *: same configuration as MSM-200 16x4 [31], with 8 attention heads in the decoder instead of 6. For model parameter counts, refer to Appendix E

Model	PT-Data	BO	CD	ESC-50	LC	Mri-S	Mri-T	NS-5h	SC-5h	F50K	VL	$s(m)$
Naive Baselines												
HEAR-Naive [48]	-	52.6±2.4	30.9±0.8	5.8±0.2	33.5±1.1	38.0±1.3	36.4±1.9	18.6±4.4	8.5±0.4	7.1±0.2	11.2±0.5	5.0±0.7
Supervised												
PaSST-base [53]	AS-5k	94.9±0.5	61.0±0.2	**94.8±0.3**	60.1±0.2	96.5±0.1	87.6±0.6	23.3±0.9	66.6±1.4	**64.2±0.1**	25.5±0.8	73.5±0.4
SSL												
W2V2-base [10]	LS-960	74.0±1.0	46.4±0.3	31.1±0.4	51.2±0.2	77.3±0.2	55.1±0.3	7.4±0.8	90.8±0.3	18.1±0.1	35.5±0.8	43.1±0.2
W2V2-large [10]	VP-100k	93.1±0.7	66.9±0.4	60.1±0.5	62.4±0.3	93.9±0.1	77.4±0.2	42.0±1.0	87.6±0.5	34.2±0.1	53.6±1.0	74.0±0.4
WavLM-base [21]	LS-960	89.4±0.7	56.3±0.2	46.6±0.4	63.2±0.3	95.1±0.1	83.4±0.2	37.3±0.8	57.2±0.8	29.9±0.1	22.6±0.6	60.5±0.2
WavLM-large [21]	Mix-94k	96.4±0.5	57.2±0.4	47.9±0.4	61.1±0.3	96.8±0.1	89.5±0.1	53.7±0.5	46.2±0.8	29.0±0.1	23.7±0.9	64.0±0.2
HuBERT-base [14]	LS-960	92.1±0.6	70.8±0.2	57.8±0.6	56.5±0.3	94.4±0.1	84.9±0.3	19.4±0.7	**93.2±0.1**	32.3±0.1	61.8±0.6	72.5±0.2
HuBERT-large [14]	LL-60k	94.1±0.7	70.7±0.1	60.3±0.4	59.9±0.2	95.3±0.1	83.5±0.3	19.3±0.8	83.2±0.7	31.5±0.1	**66.1±0.9**	73.4±0.3
SSaST-base [22]	AS+LS	93.4±0.9	56.5±0.2	68.4±0.4	60.7±0.3	96.7±0.1	96.3±0.1	66.8±0.7	53.5±1.3	38.2±0.1	28.5±0.9	71.7±0.2
BEATs-iter3 [47]	AS-5k	94.0±0.8	67.3±0.2	83.7±0.6	68.0±0.2	94.7±0.1	95.8±0.1	69.4±0.8	85.2±0.3	53.6±0.2	38.5±1.0	85.7±0.3
MAE based												
AudioMAE [37]	AS-5k	93.7±0.6	68.2±0.2	60.6±0.4	42.2±0.2	89.2±0.2	86.6±0.2	64.5±0.8	28.6±1.5	37.9±0.1	29.7±1.0	62.9±0.3
MAE-B-4x16-4l*	AS-5k	96.2±0.3	72.2±0.2	80.9±0.4	67.3±0.3	97.4±0.1	98.3±0.1	68.3±0.4	89.4±0.3	50.4±0.1	43.1±0.9	88.1±0.2
MAE-B-5x5-4l	AS-5k	96.0±0.4	70.9±0.2	80.9±0.4	67.6±0.4	**97.6±0.1**	98.4±0.0	69.3±0.4	88.4±0.3	49.3±0.2	37.7±0.6	86.8±0.2
MAE-L-4x16-8l	AS-5k	96.1±0.4	73.8±0.1	81.6±0.3	68.5±0.2	97.6±0.1	98.3±0.0	69.0±0.5	91.2±0.2	51.8±0.1	46.9±0.8	90.0±0.2
Proposed												
MW-MAE-B-4x16-4l	AS-5k	96.0±0.5	73.1±0.3	81.2±0.4	68.8±0.2	97.4±0.1	97.9±0.1	69.3±0.6	90.9±0.2	51.2±0.2	44.2±0.9	89.2±0.2
MW-MAE-B-5x5-4l	AS-5k	**96.6±0.4**	73.8±0.4	82.0±0.3	**70.1±0.4**	97.5±0.1	**98.3±0.1**	72.9±0.5	91.7±0.2	51.3±0.1	44.2±0.6	90.6±0.1
MW-MAE-L-4x16-8l	AS-5k	95.9±0.3	**76.1±0.2**	83.6±0.3	69.7±0.3	97.4±0.0	98.2±0.1	71.2±0.7	93.0±0.1	53.5±0.1	51.9±0.7	**92.6±0.2**

and to train a shallow MLP classifier with a single hidden layer with 1024 neurons for each task in a reproducible manner. Experiments are repeated with at least ten random seeds for each task, resulting in 100 experiments for every evaluated representation.

4.3 Comparison with Existing Works

Table 1 shows how MW-MAE fares against recent audio representations. The highlighted model configurations that we pre-trained from scratch on AudioSet have the following naming convention: the first substring shows the type of MAE (vanilla or proposed MW-MAE), followed by a single alphabet denoting ViT Encoder configuration. This is followed by the patch size used, and finally, the depth of the decoder. It's worth noting that while embedding sizes of MAE and corresponding MW-MAE configurations are the same, the embedding sizes of other methods can be different. This is inline with the current consensus of evaluating self-supervised representations in the audio domain [51, 48]. MW-MAE configurations outperform all other comparable MAEs, with the largest "MW-MAE-L-16x4-8l" configuration outperforming all the methods in overall performance (92.6±0.2). MW-MAEs also outperform AudioMAE with standard shifting window based attention, as well as BEATs-iter3, which is the pretrained representation obtained after 3 stages of self-distilled learning as proposed by [47]. MW-MAEs perform exceptionally well on pitch perception (NS-5h), while achieving performance on-par with speech specific representations such as WavLM, HuBERT and Wav2Vec2 (denoted W2V2) for Keyword spotting (SC-5h). Perhaps more surprisingly, they outperform speech representations trained on much larger training sets on the emotion recognition (CREMA-D) as well as speaker count classification (LibriCount) tasks. While PaSST, which is a recent state-of-the-art approach for training supervised transformers on AudioSet, outperforms every model on ESC-50 and FSD50K tasks, the overall performance of the proposed approach is significantly better. Overall, the proposed MW-MAEs learn a better general-purpose audio representation than standard MAEs, generalizing well to several audio domains and demonstrating excellent overall performance in comparison to recent audio representations.

4.4 Key Model Characteristics

We conduct several experiments to examine key differences between MAE and the proposed MW-MAE. While we have only reported overall score $s(m)$, detailed results for all these experiments can be found in Section F.

MW-MHA in the encoder: As previously mentioned, adding MW-MHA to the encoder block does not improve downstream performance. Further, we also investigate the impact of linear probing as well as fine-tuning the entire encoder stack for in-domain classification on AudioSet-20k balanced subset. No data augmentations were used. As evident from Table 2, when compared with including MW-MHA blocks in the decoder only, there is no performance benefit to adding MW-MHA blocks to the encoder for neither downstream performance, nor for in-domain linear probe on AudioSet-20k. However, when fine-tuning the entire encoder stack, adding MW-MHA blocks to the encoder provides a slight improvement and is worth considering.

Model	Downstream $s(m)$	Linear Probe (mAP)	Fine-tuning (mAP)
MAE Base	88.1±0.2	18.9±0.0	23.8±0.1
MW-MAE Base (decoder only)	89.2±0.2	20.2±0.0	23.9±0.1
MW-MAE Base (enc+dec)	89.1±0.3	20.2±0.0	24.2±0.1

Table 2: Performance impact of MW-MHA module placement

Performance impact of various patch sizes: In an MAE, the patch embedding layer generates non-overlapping patches from the input. Thus, the size of the patch governs the number of patches as well as the time-frequency resolution that the transformer layers work at, making it an important hyperparameter to investigate.

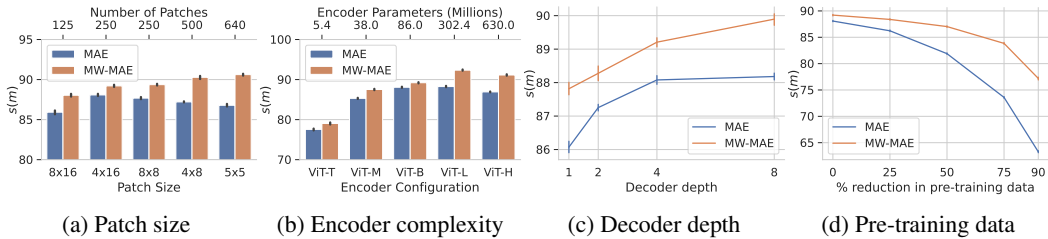(a) Patch size (b) Encoder complexity (c) Decoder depth (d) Pre-training data

Figure 2: Ablation experiments comparing standard MAE v/s proposed MW-MAE at different patch sizes (a), encoder complexity (b), decoder depth (c) as well as amount of pre-training data used (d). $s(m)$ is the proposed overall score (Sec 4). Detailed results can be found in Appendix F.

Figure 2a shows how different patch sizes affect downstream performance. The proposed MW-MAE model, with an overall score of 90.6±0.1, outperforms standard MAE for every patch size for identical decoder configurations. It's also worth noting that MAE performance degrades as we decrease the patch size beyond 4×16, whereas MW-MAE performance continues to improve. These observations show that the proposed MW-MAE adapts better to varying patch sizes and time-frequency resolutions, while scaling well with increasing number of patches.

Encoder size: As shown in Figure 2b, we investigate how encoders of five different complexities affect overall performance. All the trained models have the same decoder configuration (384 neurons, *depth*=4, *h*=8). With an overall score of 89.2±0.2, MW-MAE with the ViT-Base encoder performs better than MAEs with encoders of any size in this experiment. The most prominent performance gap is observed for the ViT-Large setting, where MAE and MW-MAE attain overall scores of 88.2±0.2 and 92.3±0.2, respectively. The drop in performance for the ViT-Huge encoder for both MAEs and MW-MAEs suggests possible overfitting.

Decoder depth: In Figure 2c, we show how increasing decoder complexity by increasing decoder depth affects overall performance. As expected, increasing decoder depth improves performance for both methods. For decoder *depth*=8, MW-MAE (89.9±0.2) outperforms MAE (88.2±0.1) by a considerable margin in overall performance. We also observed that with an overall score of 88.3±0.2, MW-MAE with *depth*=2 performs on par with MAEs with up to 4 decoder blocks. This observation complements the inherent asymmetric nature of Masked Autoencoders, and thus the proposed MW-MAE performs favourably in terms of complexity and scalability.

Pre-training data: Finally, Figure 2d depicts how performance varies as we reduce the amount of data used for pre-training. Overall, performance for both the MAE and the proposed MW-MAE methods continues to decrease monotonically as we remove more and more data. However, the

(a) Attention entropies in encoder blocks   (b) Mean attention distances for the select transformer blocks.
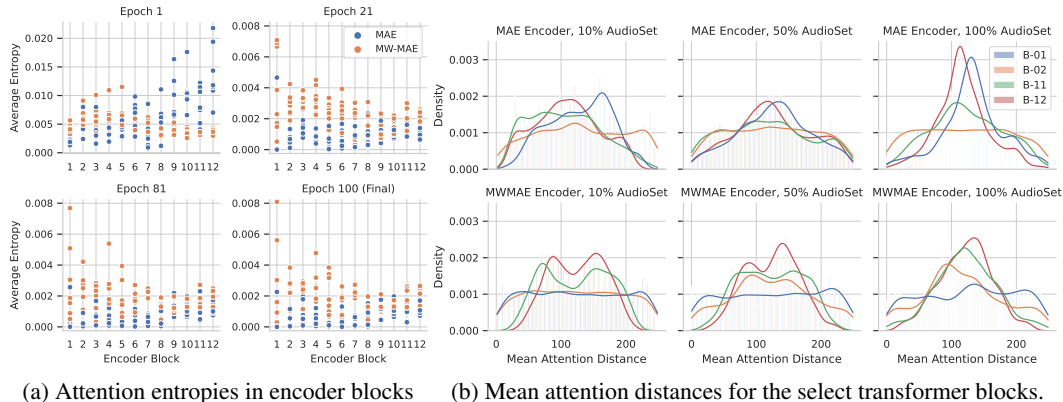
Figure 3: Investigating MAE and MW-MAE encoder attention heads. (a) depicts average entropies of encoder attention heads over the course of pretraining in a every encoder transformer block. (b) Depicts mean attention distance distributions of the first two and the last two transformer blocks at different amounts of pretraining data used.

performance loss trend for MW-MAE is much more favourable. A 90% reduction in the amount of pre-training data results in a 28.17% reduction in performance for standard MAEs (from $88.1_{\pm 0.2}$ to $63.3_{\pm 0.2}$), whereas MW-MAE only suffers a 13.5% drop in performance (from $89.2_{\pm 0.2}$ to $77.2_{\pm 0.3}$). Thus, we conclude that the proposed MW-MAEs are more adept at handling low-data scenarios in comparison to standard MAEs.

5 Exploratory Analysis

5.1 Inspecting encoder attention heads

Analyzing attention entropies: We first analyze individual attention heads in a ViT-Medium encoder (*depth*=12, h=8). Figure 3a shows scatter plots of average entropies of individual encoder attention heads computed over the entire NSynth Pitch 5h validation set on a block-by-block basis at different stages during pre-training. It's worth noting that the higher the entropy, the more global the attention, with lower attention mass spent on closer tokens [54], and thus, a higher variance in entropies of individual attention heads highlights more spread out local and global attention. In the early epochs, MAE encoders actually have higher variance in entropy distribution, especially in the latter transformer layers. As pretraining goes on, interestingly, this effect is reversed, and the attention heads in the MW-MAE encoder now start converging towards high entropy variance configurations in the early layers.

Analyzing attention distances: We analyze mean attention distances for attention heads in the first two and the last two encoder blocks. Similar to [55], we compute attention-weighted patch distances between the query patch position and the locations it attends to for each attention head, averaging it for all patches positions. This is repeated for all inputs in the FSD50K validation set. Figure 3b depicts the distribution of mean attention distance for MAE and MW-MAE encoders (base configuration) pretrained with different amounts of training data. We can observe that MW-MAE attention heads demonstrate a broader distribution of attention distances, modelling local-global attention better than the MAE encoder especially in the first two transformer blocks.

From these observations, we can conclude that in an MW-MAE, the decoder fitted with an MW-MHA can force the encoder to better capture local-global interactions even without explicit windowed attention modules, leading to improved performance.

5.2 Comparing attention feature representations through PWCCA

Several recent works have used Canonical Correlation Analysis (CCA) to compare feature representations and learning dynamics of deep neural networks [56, 57]. We use Projection Weighted CCA (PWCCA) [38], which computes a weighted mean of the CCA vectors to compare the representations
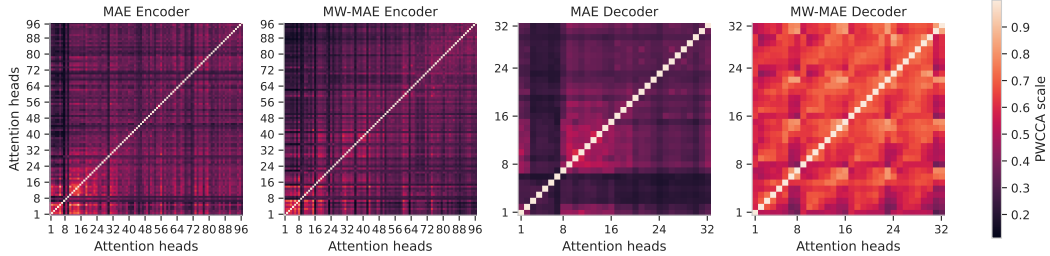
Figure 4: Comparing features learned by different attention heads in the encoder and the decoder of a standard MAE and the proposed MW-MAE using PWCCA. Each tick separates the attention heads of a transformer block from the next.

learned by individual attention heads of the encoder and the decoder in identically configured MAE and MW-MAE (ViT-M encoder: *depth*=12, *h*=8; Default decoder: 384 neurons, *depth*=4, *h*=8). MW-MAE decoder uses default attention head window sizes as specified in Sec 3.2. Figure 4 depicts correlation matrices of measured PWCCA score between attention heads. We can observe a remarkable difference in correlation between the decoders: feature representations from the MW-MAE decoder attention heads with the same window sizes are strongly correlated across decoder layers, whereas attention heads with global self-attention (7, 8, 15, 16, 23, 24, 31, 32) are the least correlated, consistent with observations made for the MAE decoder. These observations suggest a decoupling of different aspects of the feature hierarchy in the MW-MAE decoder, as attention heads of specific window sizes in each decoder block capture local information at a specific granularity, which is in line with our original hypothesis. These observations are also corroborated by decoder depth ablation experiments from Sec 4.4, where we observed that a MW-MAE with a single transformer block performs on par with MAEs fitted with up to 4 decoder blocks. Finally, the difference in correlation matrices between the encoders is much less stark, which is expected since both use standard MHA blocks.

6 Conclusion

This work presents Multi-Window Masked Autoencoder (MW-MAE) for learning general-purpose audio representations. Decoders in MW-MAEs are fitted with a novel Multi-Window Multi-Head Attention (MW-MHA) module, which learns information captured at multiple granularities of local-global context by its constituent attention heads computing self-attention over different non-overlapping windows. Empirical experiments on ten downstream tasks show that the proposed MW-MAEs consistently outperform standard MAEs in overall performance when pre-trained on the AudioSet dataset, demonstrating better scaling characteristics. Exploratory analyses highlight key differences between the attention representations learned by standard MAEs and the proposed MW-MAEs. Based on attention entropy and mean attention distance analysis, we discover that encoder attention heads in an MW-MAE better capture local-global interactions, even without explicit local-global attention modules. We also learn that attention heads of the same window size across the transformer blocks of the MW-MAE decoder are correlated, learning a decoupled feature hierarchy allowing transformers to capture relevant information at the block level, supporting our original motivation.

Acknowledgements

References

[1] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[2] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning." in *Interspeech 2019*, 2019, pp. 146–150.

[3] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[4] Y.-A. Chung, H. Tang, and J. R. Glass, "Vector-quantized autoregressive predictive coding." in *Interspeech 2020*, 2020, pp. 3760–3764.

[5] A. H. Liu, Y.-A. Chung, and J. R. Glass, "Non-autoregressive predictive coding for learning speech representations from local dependencies," in *Interspeech 2021*, 2021.

[6] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry, and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Processing Letters*, vol. 27, pp. 600–604, 2020.

[7] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3875–3879.

[8] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 371–375.

[9] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 449–12 460.

[11] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, and M. Auli, "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," in *Interspeech 2021*, 2021.

[12] A. K. Sarkar, Z.-H. Tan, H. Tang, S. Shon, and J. Glass, "Time-contrastive learning based deep bottleneck features for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1267–1279, 2019.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.   Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units." *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2021.

[15] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9653–9663.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.   Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880.

[20] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022.

[21] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[22] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 10 699–10 709.

[23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[24] C. Feichtenhofer, Y. Li, K. He *et al.*, "Masked autoencoders as spatiotemporal learners," *Advances in neural information processing systems*, vol. 35, pp. 35 946–35 958, 2022.

[25] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 668–14 678.

[26] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 594–604.

[27] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*.   Springer, 2022, pp. 604–621.

[28] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, "Multimae: Multi-modal multi-task masked autoencoders," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*.   Springer, 2022, pp. 348–367.

[29] Y. Seo, D. Hafner, H. Liu, F. Liu, S. James, K. Lee, and P. Abbeel, "Masked world models for visual control," in *Conference on Robot Learning*.   PMLR, 2023, pp. 1332–1344.

[30] A. Baade, P. Peng, and D. Harwath, "MAE-AST: Masked Autoencoding Audio Spectrogram Transformer," in *Proc. Interspeech 2022*, 2022, pp. 2438–2442.

[31] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation," in *HEAR: Holistic Evaluation of Audio Representations (NeurIPS 2021 Competition)*, ser. Proceedings of Machine Learning Research, vol. 166.   PMLR, 13–14 Dec 2022, pp. 1–24.

[32] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech 2020*.   ISCA, Oct. 2020, pp. 5036–5040.

[33] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "CSWin Transformer: A General Vision Transformer Backbone With Cross-Shaped Windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2022, pp. 12 124–12 134.

[34] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6824–6835.

[35] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4804–4814.

[36] W. Zhu and M. Omar, "Multiscale audio spectrogram transformer for efficient audio classification," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.   IEEE, 2023, pp. 1–5.

[37] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[38] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems*, vol. 31.   Curran Associates, Inc., 2018.

[39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML 2020: 37th International Conference on Machine Learning*, vol. 1, 2020, pp. 1597–1607.

[40] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[41] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 271–21 284.

[42] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Self-supervised learning for general-purpose audio representation," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[43] G. Elbanna, N. Scheidwasser-Clow, M. Kegler, P. Beckmann, K. El Hajal, and M. Cernak, "Byol-s: Learning self-supervised speech representations by bootstrapping," in *HEAR: Holistic Evaluation of Audio Representations*. PMLR, 2022, pp. 25–47.

[44] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations*, 2019.

[45] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6419–6423.

[46] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 1298–1312.

[47] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 5178–5193.

[48] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, "HEAR: Holistic Evaluation of Audio Representations," in *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, Jul. 2022, pp. 125–145, iSSN: 2640-3498.

[49] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[50] J. F. Gemmeke, D. P. 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[51] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. Jeff Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K.-t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[52] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[53] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Proc. Interspeech 2022*, 2022, pp. 2753–2757.

[54] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does BERT Look at? An Analysis of BERT's Attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286.

[55] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[56] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[57] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-Wise Analysis of a Self-Supervised Speech Representation Model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2021, pp. 914–921.

[58] M. Tian, A. Srinivasamurthy, M. Sandler, and X. Serra, "A study of instrument-wise onset detection in beijing opera percussion ensembles," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2159–2163.

[59] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.

[60] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[61] F.-R. Stöter, S. Chakrabarty, B. Edler, and E. A. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 436–440.

[62] F.-R. Stöter, S. Chakrabarty, E. Habets, and B. Edler, "Libricount, a dataset for speaker count estimation," April 2018.

[63] A. Anantapadmanabhan, A. Bellur, and H. A. Murthy, "Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 181–185.

[64] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," 2017.

[65] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018.

[66] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[67] B. Kim, M. Ghei, B. Pardo, and Z. Duan, "Vocal imitation set: a dataset of vocally imitated sound events using the audioset ontology." in *DCASE*, 2018, pp. 148–152.

Appendix

A Multi-Window Multi-Head Attention

```python
def WinAttention(Q, K, V, win_i):
  n, d_k = Q.shape[-2:]
  # partition inputs along patch dimension
  # into non-overlapping windows
  Q = Q.reshape(-1, win_i, d_k)
  K = K.reshape(-1, win_i, d_k)
  V = V.reshape(-1, win_i, d_k)
  # compute self-attention
  X = softmax(Q.(K.transpose()) / sqrt(d_k)).V
  # reshape results
  X = X.reshape(-1, n, d_k)
  return X
```

Figure 5: Pseudocode for WinAttention

B More about downstream tasks

Table 3: Overview of tasks for downstream evaluation. All these tasks are a part of the HEAR [48] benchmark.

Short Hand	Name	Description	Size (in Hours)	Metric
BO	Beijing Opera [58, 48]	Classifying percussion instruments	0.3	Accuracy
CD	Crema-D [59]	Emotion Recognition	~ 10	Accuracy
ESC-50	ESC-50 [60]	Environmental Sound Classification	2.77	Accuracy
LC	LibriCount [61, 62]	Speaker Count Identification, Simulated Cocktail Party	~ 8	Accuracy
Mri-S	Mridangam Stroke [63]	Stroke classification in pitched percussion instruments	1.57	Accuracy
Mri-T	Mridangam Tonic [63]	Tonic classification in pitched percussion instruments	1.57	Accuracy
NS-5h	NSynth Pitch 5h [48, 64]	88-way Pitch Classification, reduced training subset	~ 5.5	Accuracy
SC-5h	Speech Commands 5h [48, 65]	Keyword Spotting, reduced training subset	~ 6.5	Accuracy
F50K	FSD50K [66]	Multilabel, large scale Audio Tagging	~ 100	mAP
VL	VoxLingua107 Top10 [48, 67]	Spoken language identification	5	Accuracy

The following is our reasoning behind excluding the other tasks from the HEAR benchmark suite:

1. **Nsynth-Pitch 50hr and Speech Commands Full** because we already use the smaller subsets.

2. **Gunshot Triangulation:** Gunshot is an event in both AudioSet and FSD50k ontology, and is thus redundant.

3. **GTZAN Music Speech:** FSD50k already has music and speech labels, and the model performance correlation study in the HEAR paper [48] shows high correlation with FSD50k.

4. **GTZAN Genre:** highly correlated results with FSD50K and ESC-50 (surprisingly) as per [48]

5. **Vocal Imitations:** high correlation with LibriCount [48].

6. **Bee Hive state Classification:** large runtime costs, niche task.

7. **MAESTRO 5hr and DCASE 2016 Task 2:** significant complexity (storage, runtime, timestep based evaluation).

C Experimental Details and Hyperparameters

In this section, we provide additional experimental details. Apart from AudioSet, all other datasets are obtained directly from the HEAR [3], where they are pre-processed to 16000 Hz and distributed in a standard format.

Similar to [37], our effective learning rate (lr_{eff}) depends on the base learning rate (lr_{base}) and the batch size as follows: $lr_{\text{eff}} = lr_{\text{base}} * \frac{\text{batch size}}{256}$. In early experiments, we did not find strong augmentations at pre-training time to improve downstream performance, hence no augmentations

---

[3]https://hearbenchmark.com/hear-tasks.html

are used. For more details, refer to Table 4. As previously mentioned, hear-eval-kit[4] was used for downstream experiments, and along with the details provided here should allow for consistent, reproducible downstream experimentation.

Table 4: **Pre-training (PT) and Downstream (FT) hyperparameters**. [*]: For ViT-L and ViT-H based models, smallest batch size that didn't give OOM was used.

Configuration	AS-5k Pre-training	Downstream
Optimizer	AdamW	Adam
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.95$
Weight decay	0.05	N/A
Base learning rate	0.000015	0.0001
Learning rate schedule	linear-warmup + cosine decay	fixed
Minimum learning rate	0.0	0.0001
Dropout	0.	0.25
Warm-up epochs	10	N/A
Epochs	100	500
Early Stopping	N/A	20
Batch size	1024[*]	1024
Accelerators	8x TPU-v3 cores	1 Nvidia-A40

D Additional modality tested: ImageNet

Given the in-depth ablations and exploratory analysis, as well as resource constraints, we couldn't dive deeper into full scale testing of an additional modality given. However, as a proof of concept, we pre-trained MAE and MW-MAEs with ViT-Medium encoder on ImageNet for 100 epochs, followed by evaluating linear probe performance (training for 50 epochs).

Table 5: Proof of concept ImageNet experiments. Pre-training was done only for 100 epochs. Validation accuracy denotes linear probe performance on the ImageNet validation set.

Model	Params	Validation Accuracy
MAE Medium Encoder	38 M	29.0±0.0
MW-MAE Medium Encoder	38 M	29.8±0.0

---

[4]`https://github.com/hearbenchmark/hear-eval-kit`

E   Parameter count, averages and overall scores

Table 6: Models, number of parameters, plain average scores and overall scores of models omitted from Table 1 due to space constraints.

Model	# Params	Average	$s(m)$
HEAR-Naive	-	24.3±0.5	5.0±0.7
PaSST-base	86 M	67.5±0.3	73.5±0.4
SSL			
Wav2Vec2-base	94.4 M	48.7±0.1	43.1±0.2
Wav2Vec2-large	315.4 M	67.1±0.2	74.0±0.4
WavLM-base	94.4 M	58.1±0.1	60.5±0.2
WavLM-large	315.4 M	60.1±0.1	64.0±0.2
HuBERT-base	94.4 M	66.3±0.1	72.5±0.2
HuBERT-large	315.4 M	66.4±0.2	73.4±0.3
SSaST-base	89 M	65.9±0.1	71.7±0.2
BEATs-Iter3	90 M	75.0±0.2	85.7±0.3
MAE based			
AudioMAE	86.0 M	60.1±0.2	62.9±0.3
MAE-B-4x16-4l	86.0 M	76.4±0.1	88.1±0.2
MAE-B-5x5-4l	86.0 M	75.6±0.1	86.8±0.2
MAE-L-4x16-8l	302.4 M	77.5±0.1	90.0±0.2
Proposed			
MW-MAE-B-4x16-4l	86.0 M	77.0±0.1	89.2±0.2
MW-MAE-B-5x5-4l	86.0 M	77.8±0.1	90.6±0.1
MW-MAE-L-4x16-8l	302.4 M	79.1±0.1	92.6±0.2

F   Detailed Ablation Results

Table 7: Results from Patch size ablation experiments. ViT-B encoder was used for all experiments. n denotes total number of patches, and h denotes the number of attention heads in each decoder transformer block.

Model	BO	CD	ESC-50	LC	Mri-S	Mri-T	NS-5h	SC-5h	F50K	VL	$s(m)$
Patch Size=(8×16), n=125, h=4											
MAE	94.9±0.8	70.2±0.3	80.4±0.5	66.0±0.3	97.4±0.1	97.7±0.1	65.9±0.7	88.9±0.5	49.4±0.1	40.6±0.5	85.9±0.3
MW-MAE	95.9±0.5	72.3±0.2	81.2±0.3	68.4±0.3	97.3±0.1	97.8±0.1	67.4±0.8	90.0±0.3	50.8±0.1	41.9±0.5	88.0±0.2
Patch Size=(4×16), n=250, h=8											
MAE	96.2±0.3	72.2±0.2	80.9±0.4	67.3±0.3	97.4±0.1	98.3±0.1	68.3±0.4	89.4±0.3	50.4±0.1	43.1±0.9	88.1±0.2
MW-MAE	96.0±0.5	73.1±0.3	81.2±0.4	68.8±0.2	97.4±0.1	97.9±0.1	69.3±0.6	90.9±0.2	51.2±0.2	44.2±0.9	89.2±0.2
Patch Size=(8×8), n=250, h=8											
MAE	96.1±0.6	72.5±0.2	81.3±0.2	66.0±0.3	97.5±0.1	98.1±0.0	68.5±0.7	89.5±0.4	50.2±0.1	42.3±0.5	87.7±0.2
MW-MAE	96.3±0.4	73.0±0.1	82.6±0.3	69.3±0.3	97.5±0.1	98.1±0.1	70.3±0.8	90.5±0.1	51.4±0.1	42.3±0.5	89.4±0.1
Patch Size=(4×8), n=500, h=12											
MAE	96.7±0.2	71.3±0.3	79.0±0.4	67.8±0.3	97.7±0.0	98.5±0.0	68.7±0.4	89.0±0.4	49.8±0.2	39.2±0.7	87.2±0.1
MW-MAE	95.6±0.7	74.1±0.2	81.9±0.3	70.1±0.3	97.6±0.1	98.2±0.1	72.0±0.7	91.2±0.3	51.6±0.1	44.0±0.8	90.3±0.2
Patch Size=(5×5), n=640, h=16											
MAE	96.0±0.4	70.9±0.2	80.9±0.4	67.6±0.4	97.6±0.1	98.4±0.0	69.3±0.4	88.4±0.3	49.3±0.2	37.7±0.6	86.8±0.2
MW-MAE	96.6±0.4	73.8±0.4	82.0±0.3	70.1±0.4	97.5±0.1	98.3±0.1	72.9±0.5	91.7±0.2	51.3±0.1	44.2±0.6	90.6±0.1

Table 8: Effect of encoder size on performance. Patch size of 4×16 was used for all experiments.

Model	BO	CD	ESC-50	LC	Mri-S	Mri-T	NS-5h	SC-5h	F50K	VL	$s(m)$
Encoder=ViT-T											
MAE	95.6±0.5	63.2±0.2	70.1±0.5	64.6±0.3	97.1±0.1	97.4±0.1	66.4±0.7	74.3±0.8	41.6±0.1	26.4±0.6	77.6±0.3
MW-MAE	93.3±1.0	64.4±0.2	71.9±0.5	65.5±0.3	97.1±0.1	97.6±0.1	68.1±0.4	77.0±0.6	43.4±0.1	28.6±1.1	79.0±0.3
Encoder=ViT-M											
MAE	95.2±0.7	69.5±0.2	77.8±0.3	67.4±0.3	97.4±0.0	98.0±0.1	66.6±0.7	88.0±0.4	48.1±0.1	38.3±0.8	85.3±0.2
MW-MAE	95.9±0.3	71.8±0.3	80.3±0.4	69.7±0.1	97.2±0.1	97.8±0.1	68.1±0.5	88.8±0.6	49.6±0.1	39.8±0.8	87.5±0.2
Encoder=ViT-B											
MAE	96.2±0.3	72.2±0.2	80.9±0.4	67.3±0.3	97.4±0.1	98.3±0.1	68.3±0.4	89.4±0.3	50.4±0.1	43.1±0.9	88.1±0.2
MW-MAE	96.0±0.5	73.1±0.3	81.2±0.4	68.8±0.2	97.4±0.1	97.9±0.1	69.3±0.6	90.9±0.2	51.2±0.2	44.2±0.9	89.2±0.2
Encoder=ViT-L											
MAE	95.8±0.6	72.4±0.1	79.7±0.3	66.8±0.4	97.5±0.1	98.2±0.1	69.5±0.6	90.9±0.2	50.7±0.1	43.6±0.4	88.3±0.2
MW-MAE	95.7±0.5	75.5±0.2	82.5±0.5	70.1±0.3	97.4±0.0	98.1±0.1	70.7±0.6	93.2±0.1	53.3±0.1	51.9±0.8	92.3±0.2
Encoder=ViT-H											
MAE	96.8±0.2	71.1±0.2	78.3±0.4	67.1±0.2	97.5±0.0	98.5±0.0	67.6±0.6	89.6±0.1	49.5±0.2	40.0±0.7	86.9±0.1
MW-MAE	96.8±0.2	74.8±0.1	81.6±0.4	69.5±0.4	97.4±0.0	98.2±0.1	70.8±0.5	92.4±0.2	52.1±0.1	47.5±0.6	91.1±0.2

Table 9: Effect of decoder depth on downstream performance. ViT-B encoder, patch size of 4×16 were used for each experiment.

Model	BO	CD	ESC-50	LC	Mri-S	Mri-T	NS-5h	SC-5h	F50K	VL	$s(m)$
depth=1											
MAE	96.4±0.2	69.8±0.3	78.9±0.3	67.4±0.3	97.4±0.1	97.9±0.1	66.4±0.8	88.5±0.2	49.4±0.2	39.0±1.1	86.1±0.2
MW-MAE	96.6±0.5	72.4±0.2	79.0±0.4	68.7±0.3	97.5±0.1	98.0±0.1	68.8±0.5	90.2±0.3	50.6±0.1	39.1±0.8	87.8±0.2
depth=2											
MAE	96.8±0.3	71.3±0.3	78.8±0.2	68.8±0.2	97.4±0.1	98.2±0.0	67.2±0.6	90.0±0.2	49.6±0.2	39.4±0.7	87.3±0.1
MW-MAE	96.0±0.7	73.1±0.2	79.4±0.3	69.2±0.3	97.4±0.1	98.2±0.1	69.0±0.6	90.6±0.2	50.7±0.2	40.1±0.6	88.3±0.3
depth=4											
MAE	96.2±0.3	72.2±0.2	80.9±0.4	67.3±0.3	97.4±0.1	98.3±0.1	68.3±0.4	89.4±0.3	50.4±0.1	43.1±0.9	88.1±0.2
MW-MAE	96.0±0.5	73.1±0.3	81.2±0.4	68.8±0.2	97.4±0.1	97.9±0.1	69.3±0.6	90.9±0.2	51.2±0.2	44.2±0.9	89.2±0.2
depth=8											
MAE	96.3±0.3	71.7±0.3	81.6±0.4	67.4±0.3	97.4±0.0	98.1±0.1	67.8±0.7	89.9±0.2	50.8±0.2	43.4±0.6	88.2±0.1
MW-MAE	96.2±0.5	73.2±0.2	82.2±0.4	69.7±0.3	97.3±0.0	98.1±0.1	69.4±0.5	91.3±0.2	52.0±0.2	44.7±0.8	89.9±0.2

Table 10: Amount of pre-training dataset used v/s downstream performance.

Model	BO	CD	ESC-50	LC	Mri-S	Mri-T	NS-5h	SC-5h	F50K	VL	$s(m)$
10% of AS-5k											
MAE	93.6±0.7	51.3±0.2	49.5±0.3	48.4±0.4	97.1±0.1	96.4±0.1	61.1±0.7	70.4±0.9	29.7±0.2	17.3±0.5	63.3±0.2
MW-MAE	94.1±0.3	63.9±0.3	67.1±0.3	60.5±0.2	97.3±0.1	97.6±0.0	64.4±0.5	82.0±0.4	40.9±0.2	30.1±1.1	77.2±0.3
25% of AS-5k											
MAE	96.2±0.6	57.5±0.3	64.9±0.4	56.9±0.3	97.4±0.1	97.5±0.1	65.0±0.6	79.3±0.4	39.2±0.1	24.2±0.7	73.6±0.2
MW-MAE	96.1±0.5	68.0±0.2	75.5±0.4	67.2±0.3	97.3±0.1	98.0±0.1	65.9±0.4	86.5±0.2	46.4±0.1	35.7±0.6	83.8±0.2
50% of AS-5k											
MAE	97.2±0.3	65.5±0.3	74.1±0.3	64.3±0.3	97.5±0.1	98.1±0.1	67.0±0.6	85.3±0.6	45.1±0.1	32.4±0.8	81.9±0.2
MW-MAE	95.9±0.5	70.9±0.2	79.1±0.3	69.1±0.4	97.4±0.1	98.1±0.1	68.4±0.7	88.5±0.2	49.1±0.1	39.5±0.5	87.0±0.2
75% of AS-5k											
MAE	95.3±0.5	70.2±0.2	79.0±0.3	67.4±0.2	97.4±0.1	98.1±0.1	67.4±0.6	88.8±0.3	49.2±0.1	39.5±0.7	86.2±0.2
MW-MAE	96.0±0.5	72.6±0.3	80.5±0.4	69.5±0.3	97.4±0.1	97.9±0.1	68.3±0.4	89.9±0.2	50.5±0.1	41.7±0.8	88.4±0.2
100% of AS-5k											
MAE	96.2±0.3	72.2±0.2	80.9±0.4	67.3±0.3	97.4±0.1	98.3±0.1	68.3±0.4	89.4±0.3	50.4±0.1	43.1±0.9	88.1±0.2
MW-MAE	96.0±0.5	73.1±0.3	81.2±0.4	68.8±0.2	97.4±0.1	97.9±0.1	69.3±0.6	90.9±0.2	51.2±0.2	44.2±0.9	89.2±0.2

G Limitations

The direct limitations of our work are:

1. Pre-training data scale: As opposed to text corpus used in NLP [13] as well as speech representations [10, 14], AudioSet is several order of magnitudes smaller. While MW-MAEs demonstrate good performance characteristics in low-data scenarios, analysis on larger scales of data is definitely warranted.

2. Computational demands: transformer based models are computationally expensive to train, and despite their favourable generalization characteristics, MW-MAEs are no different. MW-MAEs and as well as previous works [31, 37] have showed the efficacy of MAEs when pretrained with AudioSet, however, training on longer duration audio data is still a challenge.

3. Runtime Overhead: While theoretically MW-MHA should be faster at runtime, the overhead of calling multiple attention heads individually results in slight slowdown as compared to optimized CUDA kernel implementations for MHA. A custom kernel for the operation should be able to improve this.