EAT: Self-Supervised Pre-Training with Efficient Audio Transformer

Wenxi Chen, **Yuzhe Liang**, **Ziyang Ma**, **Zhisheng Zheng**, **Xie Chen** *

MoE Key Lab of Artificial Intelligence, AI Institute,
X-LANCE Lab, Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
{1029713857, chenxie95}@sjtu.edu.cn

Abstract

Audio self-supervised learning (SSL) pre-training, which aims to learn good representations from unlabeled audio, has made remarkable progress. However, the extensive computational demands during pre-training pose a significant barrier to the potential application and optimization of audio SSL models. In this paper, inspired by the success of data2vec 2.0 in image modality and Audio-MAE in audio modality, we introduce **E**fficient **A**udio **T**ransformer (EAT) to further improve the effectiveness and efficiency in audio SSL. The proposed EAT adopts the bootstrap self-supervised training paradigm to the audio domain. A novel Utterance-Frame Objective (UFO) is designed to enhance the modeling capability of acoustic events. Furthermore, we reveal that the masking strategy is critical in audio SSL pre-training, and superior audio representations can be obtained with large inverse block masks. Experiment results demonstrate that EAT achieves state-of-the-art (SOTA) performance on a range of audio-related tasks, including AudioSet (AS-2M, AS-20K), ESC-50, and SPC-2, along with a significant pre-training speedup up to $\sim$15x compared to existing audio SSL models.

1 Introduction

Self-supervised learning (SSL) has emerged as a pivotal method in audio representation learning, drawing inspiration from its success in natural language processing [Devlin *et al.*, 2018; Radford *et al.*, 2018], computer vision [Chen *et al.*, 2020; He *et al.*, 2020], and speech processing [Hsu *et al.*, 2021; Chen *et al.*, 2022b; Ma *et al.*, 2023]. The strength of SSL lies in leveraging vast amounts of unlabeled data, thus enabling models to learn data features effectively.

Key to the success of SSL in the audio domain is masked autoencoder models and the bootstrap approach, celebrated for their ability to extract fruitful features from input data. Reconstruction-based methods like BERT [Devlin *et al.*, 2018] and MAE [He *et al.*, 2022] learn representations by predicting global information from limited unmasked contexts. In contrast, BYOL [Grill *et al.*, 2020] and its derivatives implement data-augmentation-based prediction tasks for continuous self-learning with online and target networks. Similar techniques have been adapted to develop audio SSL models. Models like SSAST [Gong *et al.*, 2022], MAE-AST [Baade *et al.*, 2022], Audio-MAE [Huang *et al.*, 2022] and CED [Dinkel *et al.*, 2024] concentrate on reconstructing audio spectrograms from masked patches. Others like BYOL-A [Niizumi *et al.*, 2021], ATST [Li and Li, 2022], and M2D [Niizumi *et al.*, 2023] employ self-learning based on the bootstrap framework in augmented spectrogram data to learn latent audio representations during pre-training.

Despite these developments, the expensive computational cost of pre-training remains a hurdle. Approaches like Audio-MAE attempt to enhance encoding efficiency by using a high mask ratio and feeding only unmasked patches to the encoder. However, this would necessitate a complex decoder like a SwinTransformer [Liu *et al.*, 2021], often leading to prolonged processes. Other audio SSL models aim to streamline pre-training by simplifying learning tasks. For example, in BEATs [Chen *et al.*, 2022a], using a tokenizer to discretize target features allows it to emphasize semantically rich audio tokens and thereby facilitate learning in each iteration. However, this quantitative approach may result in the loss of objective information and require more pre-training iterations.

Therefore, we introduce the **E**fficient **A**udio **T**ransformer (EAT) model, tailored for efficient learning of audio semantics and exceptional performance in downstream tasks.

EAT departs from conventional methods that focus on reconstructing audio patches or predicting discrete features. Instead, it employs a unique Utterance-Frame Objective (UFO) during pre-training, synergizing global utterance-level and local frame-level representations in its prediction task. Besides, EAT adopts a bootstrapping framework in which the student model is continuously updated with target features derived from a teacher model. This teacher model is gradually updated through an exponential moving average (EMA) technique, akin to MOCO [He *et al.*, 2020].

For the pretext task, EAT employs the Masked Language Modeling (MLM) with an 80% masking ratio, focusing on patch embeddings from downsampled audio spectrograms accompanied by fixed sinusoidal positional embeddings. Inspired by the masking method in data2vec 2.0 [Baevski *et al.*,

*Corresponding author.

2023] on image modality, EAT adopts an inverse block multi-mask technique on audio patches. This method preserves unmasked data in block units, resulting in larger regions of locality for unmasked patch embeddings and thus increasing the challenge of extracting audio semantics and predicting masked features. Additionally, the multi-mask strategy compensates for the computational cost associated with encoding the complete raw audio patches input to the teacher model during pre-training. Implementing multiple clones of masked input data (only masked parts encoded) in the student model significantly boosts data utilization efficiency.

At last, we design an asymmetric network architecture that combines a complex Transformer encoder with a lightweight CNN decoder. This setup efficiently decodes features, facilitating precise frame-level feature prediction.

With its efficient self-learning mechanism, the EAT model can adeptly acquire crucial audio features. Our experiments confirm that EAT, with significantly reduced training hours in total, achieves state-of-the-art performance on several audio and speech classification datasets, underscoring its superior generalization and learning efficiency in the audio domain.

Our contributions are summarized as follows:

- We introduce a novel Utterance-Frame Objective (UFO) during pre-training in audio SSL for learning audio latent representation. The utterance-level learning is experimented to be crucial in model pre-training.
- We adopt the inverse block multi-mask method from data2vec 2.0 with a high mask ratio on audio patches, which significantly speeds up the pre-training process in the audio bootstrap framework. Experiments show that EAT substantially outperforms previous audio SSL models in pre-training efficiency.
- We achieve SOTA results on several popular audio-related datasets. The code and models are also open-sourced to facilitate community development. [1]

2 Related Work

2.1 Bootstrap Method

The concept of the bootstrap method was initially introduced in the context of self-supervised learning by BYOL [Grill *et al.*, 2020]. The BYOL architecture incorporates a dual-component framework, consisting of a target encoder and a predictor network. The target encoder is responsible for generating representative targets, while the predictor network aims to predict these targets using an augmented version of the input. The predictor network is updated through the prediction objective, whereas the target encoder undergoes momentum updates, a concept derived from the Momentum Contrast (MoCo) method [He *et al.*, 2020]. This approach has inspired a series of subsequent self-supervised vision models, notable examples being DINO [Caron *et al.*, 2021], SimSiam [Chen and He, 2021], and MoCo v3 [Chen *et al.*, 2021].

Extending the bootstrap method to various modalities, data2vec [Baevski *et al.*, 2022] and its successor, data2vec 2.0 [Baevski *et al.*, 2023], represent significant advancements in self-supervised learning. These models utilize mask-based techniques for contrasting the pretext task, significantly enhancing pre-training efficiency. Their approach also involves regressing representations across multiple neural network layers, rather than concentrating exclusively on the top layer.

In an endeavor to embrace the potential of the bootstrap method like BYOL-A [Niizumi *et al.*, 2021] and M2D [Niizumi *et al.*, 2023], our EAT model also applies this methodology to the audio domain and aims to enhance the audio feature learning while improving the pre-training efficiency.

2.2 Self-supervised Audio Pre-training

Self-supervised learning (SSL) in the audio domain involves extensive pre-training using large volumes of unlabeled data to learn latent audio features. Typically, there are two main approaches to selecting in-domain pre-training data. The first approach is joint pre-training, which combines speech and audio data, as exemplified by models like SS-AST [Gong *et al.*, 2022] and MAE-AST [Baade *et al.*, 2022]. The second, and more prevalent approach, is to exclusively use audio data for pre-training, as seen in models such as MaskSpec [Chong *et al.*, 2023], MSM-MAE [Niizumi *et al.*, 2022], Audio-MAE [Huang *et al.*, 2022], and our EAT model.

Various methods are employed in different components of audio SSL models. For input data, models like wav2vec 2.0 [Baevski *et al.*, 2020] and data2vec process raw waveforms, whereas most others including EAT use Mel spectrograms to extract features. In terms of pretext tasks, models employing Masked Language Modeling (MLM) techniques, such as MAE-AST, Audio-MAE, and our EAT model, apply higher masking rates to audio patches. Contrastingly, models like BYOL-A [Niizumi *et al.*, 2021] and ATST [Li and Li, 2022] use augmentation techniques like mixup and random resize crop (RRC) to provide varied auditory perspectives.

The pre-training objectives also vary across models. For instance, Audio-MAE and MAE-AST use an MAE-style task, reconstructing original spectrogram patches where unmasked data predicts the masked ones. BEATs [Chen *et al.*, 2022c] employs a tokenizer for discretized semantic feature prediction. Meanwhile, models like data2vec, BYOL-A, and M2D, focus on predicting latent representations. In EAT, we have adapted the representation prediction task into the Utterance-Frame Objective (UFO) to take both global and local information in the audio spectrogram into consideration.

3 Method

EAT draws inspiration from the data2vec 2.0 [Baevski *et al.*, 2023] and Audio-MAE [Huang *et al.*, 2022] model, incorporating a blend of bootstrap and masked modeling method to effectively learn the latent representations of audio spectrogram. In this process, we devised an asymmetric network architecture that employs a standard Transformer encoder for processing visible patches (unmasked regions) and a lightweight CNN decoder for the comprehensive decoding of all features, including those at masked positions. This architecture enables rapid pre-training: complex encoding is applied to smaller data (visible patches), while a simpler decoder processes the entire data (visible features along with

[1] The code and pre-trained models is available at https://github.com/cwx-worst-one/EAT.

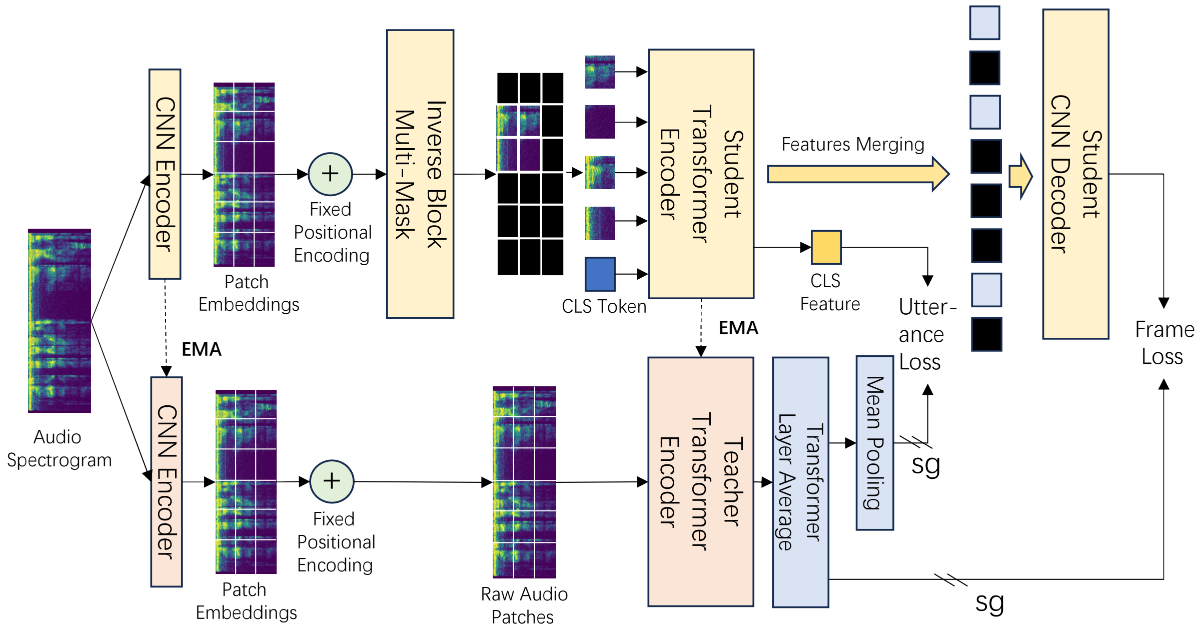

Figure 1: **Architecture of EAT in Audio Self-supervised Pre-training.** EAT first transforms the audio spectrogram into patch embeddings with a teacher and a student CNN encoder respectively. They are then separately fed into the student model via the inverse block multi-mask method and the teacher model with the same network directly. Subsequently, the generated features merged with the masked parts, are decoded using a lightweight CNN decoder. The teacher model synthesizes the average output from all Transformer layers as the target value. The utterance-level loss utilizes regression on the mean pooling values of the target values across patch dimensions, while the frame-level loss uses regression on target values at masked positions. The teacher model is updated through the EMA method, based on the learnable parameters of the student model. Notably, "sg" means stop-gradient here.

masked tokens). Furthermore, EAT distinctively combines frame-level loss, focusing on latent representation reconstruction, with utterance-level loss, targeting global representation prediction. This simple combination allows the model to adeptly capture both local nuances and overarching trends from raw audio data, significantly enhancing its performance. Figure 1 illustrates our EAT model and the details of each component, pre-training and fine-tuning are as follows.

3.1 Model Architecture

Patch Embedding with Positional Encoding. EAT is designed to operate on audio spectrograms rather than the original waveforms. To downsample audio spectrogram features, we first use padding to extend it along the time frame to a uniform length (suitable for different datasets), and then extract patch embeddings from it through a 2D convolutional layer encoder. We maintain the CNN encoder's kernel size S and stride the same to ensure the relative independence between patch embeddings by preventing overlap. Specifically, the audio spectrogram $\mathbf{X} \in R^{T\times F}$ is transformed into patch embeddings $\mathbf{X}_p \in R^{P\times E}$, where $T \times F$ represents the time and frequency dimensions of the input spectrogram and $P \times E$ denotes the patch size and embedding features dimensions, with $P = TF/S^2$ after flattening. Subsequently, 1D fixed positional encoding used in standard ViT [Dosovitskiy *et al.*, 2020] is applied to these embeddings, providing essential positional information for more effective encoding in subsequent Transformer blocks.

Utterance-Frame Objective

EAT introduces a Utterance-Frame Objective (UFO) function during pre-training, effectively merging global utterance-level and local frame-level losses in audio representation prediction. This dual-focus strategy is a significant advancement in contextualized target prediction.

The contextualized target $\mathbf{Y}_a \in R^{P\times E}$ is derived from the top-k-layers of the Transformer blocks output in the teacher model, processing complete input patch embeddings $\mathbf{Y}_r \in R^{P\times E}$. Unlike the BYOL [Grill *et al.*, 2020] method, which utilizes only the last layer's output feature as the target, EAT computes $\mathbf{Y}_a$ by averaging outputs across all Transformer layers. This approach ensures a comprehensive representation target that captures both shallow-level, raw audio features and deep-level, semantically rich latent representations.

To effectively integrate global utterance information from audio spectrograms without adding structural complexity, EAT incorporates a simple, learnable classification token (CLS token) into the student model. The multi-head self-attention mechanism of the Transformer architecture allows this CLS token $\mathbf{c} \in R^{1\times E}$ to view and access information from all unmasked patch embeddings. Then, we use the CLS feature $\mathbf{c}' \in R^{1\times E}$ from student encoder output to predict the average value of $\mathbf{Y}_a$ in patch dimension, i.e. $\mathbf{y}'_a \in R^{1\times E}$, with MSE loss. The utterance loss is calculated as follows:

$$L_u = ||\mathbf{c}' - \mathbf{y}'_a||_2^2$$

Distinctively, EAT's approach to utterance-level learning sets it apart from models like ATST-Clip [Li *et al.*, 2023]. EAT avoids additional projectors or predictors for feature transformation, directly focusing on capturing global audio features at the utterance level. This direct regression technique is experimentally shown to effectively preserve crucial information in global audio representation learning, reducing the risk of information loss during feature transformation.

For local frame-level learning in the audio patches, EAT employs the MAE [He *et al.*, 2022] method. The student encoder output representations $\mathbf{X}_d \in R^{P' \times E}$, merged with mask tokens from the original sequence, predict the average features $\mathbf{Y}_a$ at masked positions using a lightweight CNN decoder. The frame loss, also based on MSE, estimates the difference between the decoder output $\mathbf{X}_o \in R^{P'' \times E}$ and the target value $\mathbf{Y}_o \in R^{P'' \times E}$, where $P'' = T' \times F' \times M$. The frame loss is computed as:

$$L_f = \frac{1}{P''} \sum_{i=1}^{P''} ||\mathbf{X}_o - \mathbf{Y}_o||_2^2$$

Finally, the UFO loss by combining the frame-level and utterance-level losses can be given by:

$$L_{UFO} = L_f + \lambda L_u$$

λ is the hyperparameter to determine the impact of utterance loss and is found to be crucial to the overall performance of EAT as shown in Section 4.3.

Masking Strategies in Pre-training

A key to the EAT model's efficiency in learning audio representations is the masking strategy. The EAT model applies a masking rate of up to 80% to patch embeddings before encoding. This high masking rate substantially reduces the data volume processed by the Transformer, akin to the approach in MAE, thereby enhancing training speed. More importantly, it escalates the challenge of masked learning, compelling the model to decipher the essential information from the entire audio spectrogram with more limited visible input and infer the masked features during pre-training.

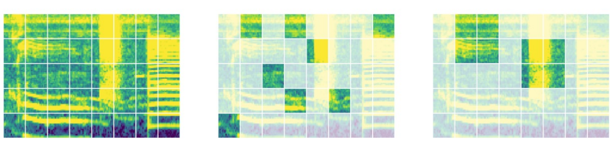

Figure 2: **Inverse Block Masking on Audio Patches.** The block size is set to 2×2 with a masking ratio of 80% in the right subfigure.

The masking method of EAT, as depicted in Figure 2, is distinct from previous audio SSL models. Instead of random masking audio patches, EAT implements inverse block masking proposed in data2vec 2.0 [Baevski *et al.*, 2023] on image modality. For a given patch embedding $\mathbf{X}_p \in R^{P \times E}$, instead of applying 1D random masking which decorrelates the time and frequency dimensions, EAT's masking reshapes $\mathbf{X}_p$ into $\mathbf{X}'_p \in R^{T' \times F' \times E}$ and applies a 2D random mask. This mask maintains correlation in both time and frequency dimensions, where $T' = T/S$ and $F' = F/S$. The process involves initially masking all patches, then iteratively preserving original parts in block size until the masked embedding count aligns with the desired masking rate. Compared to 1D random masking with the same masking ratio, it challenges EAT to concentrate on a more restricted yet focused set of fragmented audio clips for representation prediction using UFO.

In addition, EAT could be further accelerated with the multi-mask training. The teacher model, processing complete audio patch embeddings, demands greater computational resources for encoding than its student. Thus, EAT employs the multi-mask strategy to optimize efficiency, creating multiple clone-masked embeddings from the same spectrogram patch using different inverse block masking. These variants are concurrently inputted into the student model, thus amplifying data utilization via parallel computing.

3.2 Pre-training Details

EAT consists of 88M parameters during pre-training and ~86M in fine-tuning (post-CNN decoder released), aligning with the parameter scale of other standard base audio SSL models. We employ a CNN encoder with a (16,16) kernel and a stride of 16 for downsampling audio spectrograms, ensuring non-overlapping patch features extraction in the time and frequency dimensions. Both student and teacher model encoders use the 12-layer ViT-B [Dosovitskiy *et al.*, 2020] model. For faster decoding, EAT utilizes a 6-layer 2D CNN decoder with (3,3) kernels, LayerNorm [Ba *et al.*, 2016], and GELU activation [Hendrycks and Gimpel, 2016] .

During the self-supervised pre-training, the student model with parameters θ_s is updated via the UFO function. Following the general bootstrap approach, the teacher model with parameters θ_t in EAT is updated using an Exponential Moving Average (EMA) strategy. The parameter update formula [Lillicrap *et al.*, 2015] is defined as:

$$\theta_t \leftarrow \tau\theta_t + (1 - \tau)\theta_s$$

EAT employs a linearly increasing strategy for adjusting the value of τ. This approach provides the model with enhanced flexibility and randomness in the initial training stages, facilitating parameter adjustments and supporting the learning process of the student model. As training advances, τ approaches 1, leading to a more stable learning.

3.3 Fine-tuning Details

In the fine-tuning stage, EAT generates latent representations using the student Transformer encoder and replaces the original CNN student decoder with a linear layer for predicting audio categories. Additionally, we implement several data augmentation techniques to fully exploit the model's acquired comprehension of audio features from the pre-training phase.

During fine-tuning, EAT is enhanced with audio augmentations including SpecAug [Park *et al.*, 2019], mixup [Zhang *et al.*, 2017], droppath [Huang *et al.*, 2016], audio rolling, and random noise. Specifically, mixup is applied to spectrograms, aligning with EAT's pre-training focus on spectrogram-based latent representations. For classification tasks, a CLS token is used for final prediction, which shows improved performance over mean pooling methods in our experiments in Section 4.3.

Model	#Param	Pre-training Data	AS-2M mAP(%)	AS-20K mAP(%)	ESC-50 Acc(%)	SPC-2 Acc(%)
Supervised Pre-Training						
PANN [Kong *et al.*, 2020]	81M	-	43.1	27.8	83.3	61.8
PSLA [Gong *et al.*, 2021b]	14M	IN	44.4	31.9	-	96.3
AST [Gong *et al.*, 2021a]	86M	IN	45.9	34.7	88.7	98.1
MBT [Nagrani *et al.*, 2021]	86M	IN-21K	44.3	31.3	-	-
PassT [Koutini *et al.*, 2021]	86M	IN	47.1	-	96.8	-
HTS-AT [Chen *et al.*, 2022a]	31M	IN	47.1	-	97.0	98.0
Wav2CLIP [Wu *et al.*, 2022]	74M	TI+AS	-	-	86.0	-
AudioCLIP [Guzhov *et al.*, 2022]	93M	TI+AS	25.9	-	96.7	-
Self-Supervised Pre-Training						
Conformer [Srivastava *et al.*, 2022]	88M	AS	41.1	-	88.0	-
SS-AST [Gong *et al.*, 2022]	89M	AS+LS	-	31.0	88.8	98.0
MAE-AST [Baade *et al.*, 2022]	86M	AS+LS	-	30.6	90.0	97.9
MaskSpec [Chong *et al.*, 2023]	86M	AS	47.1	32.3	89.6	97.7
MSM-MAE [Niizumi *et al.*, 2022]	86M	AS	-	-	85.6	87.3
data2vec [Baevski *et al.*, 2022]	94M	AS	-	34.5	-	-
Audio-MAE [Huang *et al.*, 2022]	86M	AS	47.3	37.1	94.1	**98.3**
BEATs$_{iter1}$ [Chen *et al.*, 2022c]	90M	AS	47.9	36.0	94.0	**98.3**
BEATs$_{iter2}$ [Chen *et al.*, 2022c]	90M	AS	48.1	38.3	95.1	**98.3**
BEATs$_{iter3}$ [Chen *et al.*, 2022c]	90M	AS	48.0	38.3	95.6	**98.3**
BEATs$_{iter3+}$ [Chen *et al.*, 2022c] *	90M	AS	48.6	38.9	98.1	98.1
Ours						
EAT	88M	AS	**48.6**$_{\pm 0.05}$	**40.3**$_{\pm 0.04}$	**96.0**$_{\pm 0.06}$	**98.3**$_{\pm 0.04}$

Table 1: **Model Comparison among existing methods in audio classification tasks.** Pre-training data sources include ImageNet (IN), AudioSet (AS), and LibriSpeech (LS), while CLIP utilizes 400M text-image pairs (TI). We gray-out the methods with additional supervised training on external datasets or additional pseudo-labels. *: Models employ knowledge distillation across iterations with extra pseudo-labels.

4 Experiments

We pre-trained EAT on the AudioSet-2M (AS-2M) dataset [Gemmeke *et al.*, 2017], evaluating its performance through audio-classification fine-tuning on AS-2M, AS-20K, and the Environmental Sound Classification (ESC-50) [Piczak, 2015] datasets, as well as speech-classification fine-tuning on the Speech Commands V2 (SPC-2) [Warden, 2018] dataset.

4.1 Experimental Setups

AudioSet (AS-2M, AS-20K). AudioSet, comprising $\sim$2 million YouTube clips of 10 seconds each, spans 527 classes. In our experiment, we downloaded and processed 1,912,134 clips as the unbalanced set (AS-2M) and 20,550 as the balanced set (AS-20K), with an evaluation set of 18,884 clips. Given the multi-category nature of these clips, we employed mean Average Precision (mAP) as our test metric, which calculates the average precision across multiple classes.

Environmental Sound Classification (ESC-50). ESC-50 dataset consists of 2,000 audio clips, each five seconds long and distributed across 50 semantic classes. In our evaluation, we implemented a five-fold cross-validation method, using 400 clips for validation and the remaining for training in each fold. The evaluation metric is the average validation accuracy across five-fold experiments.

Speech Commands V2 (SPC-2). SPC-2 is a keyword-spotting task in speech recognition, comprising 35 specific speech commands. It includes 84,843 training recordings, 9,981 validation recordings, and 11,005 testing recordings, each lasting 1 second. We utilized the data split from the SUPERB [Yang *et al.*, 2021] benchmark to evaluate accuracy.

Training Details

We uniformly resampled the input waveforms to 16kHz sample rate, then transformed them into 128-dimensional Mel-frequency bands using a 25ms Hanning window with a 10ms shift. To preserve edge features during feature extraction with the CNN encoder, padding was applied to the Mel spectrogram. Additionally, the audio spectrogram patches are then normalized with a mean value of 0 and a standard deviation of 0.5, following the approach used in previous works.

EAT is pre-trained using AS-2M for 10 epochs with a batch size of 12 and a peak learning rate of 0.0005. For each clip, we created 16 clones with different inverse block masks via the multi-mask method. The cosine annealing learning strategy with warm-up steps [Loshchilov and Hutter, 2016] was employed, alongside the Adam optimizer [Loshchilov and Hutter, 2017], with β_1 and β_2 values set to 0.9 and 0.95, respectively. We distribute the training load over 4 RTX 3090 GPUs and the total training time is around 58 hours.

4.2 Main Results

Model Performance

Table 1 presents the evaluation results of EAT (with 3-run error bars) and prior models on AS-2M, AS-20K, ESC-50, and SPC-2 datasets, respectively. We categorize them into Supervised Pre-Training and Self-supervised Pre-Training models.

For fair comparison, our performance evaluation benchmark primarily focuses on Self-supervised Pre-Training models.

In the audio classification task, the EAT model achieved SOTA performance on AS-2M, AS-20K, and ESC-50 datasets. On AS-2M, EAT reached an mAP of 48.6%, surpassing the previous SOTA by 0.6%. Notably, it scored an mAP of 40.3% on the AS-20K dataset, exceeding the previous SOTA by 2.0%. In the ESC-50 dataset, EAT achieved a 96.0% accuracy, lowering the average error rate from 4.4% to 4.0%. Besides, EAT also excelled in speech classification, despite the primary focus being on audio datasets. In SPC-2, EAT achieved competitive accuracy, reaching 98.3%, consistent with previous SOTA models. These results emphasize EAT's versatility and its broad applicability across various audio and speech classification tasks.

Backbone	#Param	AS-20K	AS-2M
ViT-B	88M	40.3	48.6
ViT-L	309M	**42.0**	**49.5**

Table 2: **Experiment on Scaling the model size for EAT.**

We also experimented with scaling up the model size, using a 24-layer ViT-L as the backbone for the EAT-large model. After 20 epochs of pre-training on Audioset, the EAT-large achieved a test result of 42.0% mAP on AS-20K and 49.5% mAP on AS-2M, greatly surpassing the previous SOTA. Table 2 demonstrates that increasing the scale of the EAT model improves performance greatly.

model	epoch	hour $\times$ GPU	speedup	mAP
BEATs$_{iter3}$	342	3600	1$\times$	38.3
Audio-MAE	32	2304	1.56$\times$	37.1
EAT	**10**	**230**	**15.65**$\times$	**40.3**

Table 3: **Comparison with BEATs$_{iter3}$ and Audio-MAE on pre-training cost.** We evaluate the pre-training wall-clock time of EAT on 4 RTX 3090 GPUs in Fairseq [Ott *et al.*, 2019] and it demands around 5.8 hours for each epoch. BEATs is pre-trained on 16 Tesla V100-SXM2-32GB GPUs for around 75 hours per iteration with 114 epochs while Audio-MAE on 64 V100 GPUs for approximately 36 hours in total. All models are uniformly fine-tuned on AS-20K.

Pre-training Efficiency

The EAT model showcases exceptional efficiency during its pre-training phase compared to previous SOTA audio self-supervised learning models. As depicted in Table 3, EAT, pre-trained for just 10 epochs, achieves a total pre-training time reduction of 15.65 times compared to BEATs$_{iter3}$ and 10.02 times relative to Audio-MAE. Furthermore, as shown in Figure 3, EAT matches Audio-MAE's performance after only two epochs and surpasses BEATs$_{iter3}$ by the fifth epoch. This substantial enhancement in training efficiency greatly reduces the computational resources, easing the pre-training process for a high-performing base audio SSL model.

The efficiency gains of EAT are attributable to two key aspects. First, EAT adopts a high mask ratio of 80% during pre-training. This substantial masking implies that a significant portion of audio data is excluded before being fed to the student encoder, enhancing batch processing capacity and exploiting the benefits of parallel computing for improved efficiency. Second, our proposed Utterance-Frame Objective (UFO) function, a departure from the traditional audio spectrogram patch reconstruction objective, requires only lightweight decoding. Thus, EAT employs a lightweight CNN decoder for feature decoding and prediction, in contrast to the Transformer blocks used in models like Audio-MAE, which greatly speeds up the pre-training process.

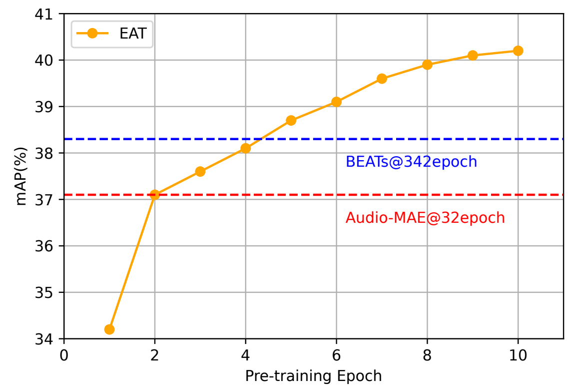

Figure 3: **Comparison with BEATs$_{iter3}$ and Audio-MAE on pre-training epoch during EAT's 10-epoch pre-training.** All models are uniformly fine-tuned on AS-20K and tested on the evaluation set.

Table 3 and Figure 3 illustrate EAT's efficiency in delivering outstanding performance with a small number of training epochs. Notably, EAT outperforms Audio-MAE (pre-trained for 32 epochs) and BEATs$_{iter3}$ (pre-trained for 342 epochs) on AS-20K after just 5 epochs of pre-training. Furthermore, after a total of 10 epochs of pre-training, EAT's mAP test score reached 40.3%, markedly outperforming these two models that underwent more extensive pre-training.

This impressive performance with fewer training epochs is largely due to the EAT's multi-mask strategy on the audio spectrogram during the utterance-frame union pre-training. By employing multiple clones with different block masking on the audio patch embeddings, EAT effectively learns to "listen" to fragmented audio from various perspectives, enabling a more comprehensive understanding and superior performance than the single-angle one. Despite the reduction in input batch size per update, this strategy significantly enhances the data utilization of each audio clip, thereby substantially improving the efficiency of the EAT model.

4.3 Ablation Study

We conduct comprehensive ablation studies to evaluate the contributions of key components in EAT. These studies evaluated different configurations of EAT, all pre-trained for 10 epochs on AS-2M and then fine-tuned on AS-20K.

Utterance-level Learning

Our experiments delved into the significance of utterance-level learning by analyzing the impact of the utterance loss

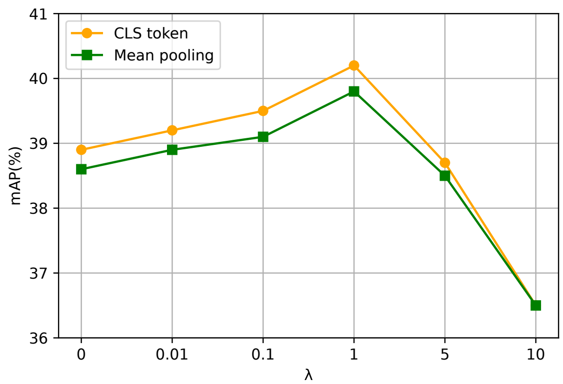

Figure 4: **Comparison on Utterance-level Loss Weight λ in Pre-training and Prediction Methods in Fine-tuning.** During fine-tuning, we compare the effect of the final prediction on using the CLS token and mean pooling over all frames.

weight λ during pre-training, as well as the effectiveness of the CLS-token-predicting method during fine-tuning.

Figure 4 illustrates that incorporating utterance loss L_u alongside frame loss L_f notably enhances the performance of EAT. Adopting a balanced approach with an utterance to frame loss weight ratio of 1:1 ($\lambda = 1$) not only provides a 1.3% increase in mAP over a model configuration with no utterance loss ($\lambda = 0$) but also shows a 1.0% improvement compared to a skewed ratio of 1:100 ($\lambda = 0.01$). However, an excessively high utterance loss weight ($\lambda = 10$) results in diminished performance, indicating that overemphasis on utterance-level learning would compromise the model's overall understanding abilities on audio clips.

Additionally, as Figure 4 shows, our experiment reveals a distinct advantage in using the CLS token for predictions over the mean pooling method during fine-tuning. While mean pooling, averaging encoder output features across the patch dimension, is commonly effective in many audio SSL models, EAT's focus on global features through increased utterance loss weight during pre-training enhances the learnable CLS token's ability to extract global features. Consequently, this approach leads to improved performance of EAT in classification tasks.

In summary, appropriately weighting the utterance loss during pre-training enhances EAT's focus on global audio spectrogram features, fostering a more comprehensive latent representation learning. Additionally, using the CLS token for prediction in fine-tuning further boosts the model's performance, leveraging these global features for improved audio classification.

Inverse Block Masking on Audio Patches

In exploring the impact of the masking strategy during pre-training, we observed notable differences in EAT's performance. Table 4 illustrates that the inverse block masking (with block size $S > 1 \times 1$) on audio patches performs better compared to the random masking ($S = 1 \times 1$). Notably, EAT configured with an increased inverse block size of $S = 5 \times 5$ attained the highest evaluation mAP of 40.3%.

Block Size	mAP(%)
1×1	37.8
2×2	39.5
3×3	39.9
4×4	40.0
5×5	**40.3**
6×6	39.8
7×7	39.8
8×8	39.8
$5 \times 5, 6 \times 4, 8 \times 3$	**40.3**

Table 4: **Comparison on different block sizes during EAT pre-training within the inverse block masking on audio patches.**

We conducted experiments with flexible block size sampling for masking, allowing the model to randomly preserve audio patches in block size like 5×5, 6×4, and 8×3 during pre-training. The outcomes were similar to using only 5×5 blocks, suggesting that block shape has a limited impact on performance. Instead, the key factors are block size and quantity in the mask. With a fixed 80% mask ratio, properly increasing the block size (and correspondingly, reducing the total number of preserved blocks) in the mask is instrumental in enhancing the model's performance. When the block size is small, numerous preserved blocks scattered across audio patches make it easier for the model to deduce masked parts, limiting its ability to deeply understand audio representations. Conversely, using sufficiently large blocks for inverse masking effectively reduces the mutual information between visible and masked audio patches, aiding the model in learning to extract features from a more constrained set of known information and predict the unknown patches.

5 Conclusion

In this paper, we propose an **E**fficient **A**udio **T**ransformer (EAT) model for effective and efficient audio-based self-supervised learning. EAT stands out by significantly expediting the pre-training process and delivering exceptional performance. Central to EAT's design is the novel use of the Utterance-Frame Objective (UFO) loss, which is proven instrumental in learning audio latent representations. The integration of utterance-level learning, enhanced by balancing its loss weight with the frame-level learning during pre-training and employing CLS-token-based prediction in fine-tuning, effectively captures global audio features. EAT achieves state-of-the-art (SOTA) results in several audio and speech classification tasks, including AudioSet, ESC-50, and SPC-2, surpassing existing base audio SSL models in overall performance. The implementation of an inverse block multi-mask method with a high mask ratio on audio spectrogram patches contributes to EAT's expedited pre-training, outpacing models like Audio-MAE and BEATs by more than tenfold in terms of time efficiency.

In the future, we aim to explore more advanced applications for EAT, focusing on complex audio understanding and generation. Additionally, we aim to investigate audio-speech joint training, delving into the interplay between these two domains using our EAT model.

Acknowledgments

We thank Sanyuan Chen for the helpful discussions and feedback. This work was supported by the National Natural Science Foundation of China (No. U23B2018, No. 62206171), and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102 and the International Cooperation Project of PCL.

References

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[Baade *et al.*, 2022] Alan Baade, Puyuan Peng, and David Harwath. MAE-AST: Masked autoencoding audio spectrogram Transformer. *arXiv preprint arXiv:2203.16691*, 2022.

[Baevski *et al.*, 2020] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Proc. NeurIPS*, 2020.

[Baevski *et al.*, 2022] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proc. ICML*, 2022.

[Baevski *et al.*, 2023] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *Proc. ICML*, 2023.

[Caron *et al.*, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, et al. Emerging properties in self-supervised vision Transformers. In *Proc. ICCV*, 2021.

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. CVPR*, 2021.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2021] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision Transformers. In *Proc. ICCV*, 2021.

[Chen *et al.*, 2022a] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. HTS-AT: A hierarchical token-semantic audio Transformer for sound classification and detection. In *Proc. ICASSP*. IEEE, 2022.

[Chen *et al.*, 2022b] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. In *Proc. JSTSP*, 2022.

[Chen *et al.*, 2022c] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. BEATs: Audio pre-training with acoustic tokenizers. *Proc. ICML*, 2022.

[Chong *et al.*, 2023] Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng. Masked spectrogram prediction for self-supervised audio pre-training. In *Proc. ICASSP*. IEEE, 2023.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dinkel *et al.*, 2024] Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, and Yujun Wang. CED: Consistent ensemble distillation for audio tagging. In *Proc. ICASSP*, 2024.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Gemmeke *et al.*, 2017] Jort Gemmeke, Daniel Ellis, Dylan Freedman, Aren Jansen, et al. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. ICASSP*. IEEE, 2017.

[Gong *et al.*, 2021a] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram Transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[Gong *et al.*, 2021b] Yuan Gong, Yu-An Chung, and James Glass. PSLA: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021.

[Gong *et al.*, 2022] Yuan Gong, Cheng Lai, Yu-An Chung, and James Glass. SSAST: Self-supervised audio spectrogram Transformer. In *Proc. AAAI*, 2022.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin Tallec, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Proc. NeurIPS*, 2020.

[Guzhov *et al.*, 2022] Andrey Guzhov, Federico Raue, Jorn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *Proc. ICASSP*. IEEE, 2022.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. CVPR*, 2022.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[Hsu *et al.*, 2021] 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. In *Proc. TASLP*, 2021.

[Huang *et al.*, 2016] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Proc. ECCV*. Springer, 2016.

[Huang *et al.*, 2022] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, et al. Masked autoencoders that listen. In *Proc. NeurIPS*, 2022.

[Kong *et al.*, 2020] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark Plumbley. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[Koutini *et al.*, 2021] Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio Transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

[Li and Li, 2022] Xian Li and Xiaofei Li. ATST: Audio representation learning with teacher-student Transformer. *arXiv preprint arXiv:2204.12076*, 2022.

[Li *et al.*, 2023] Xian Li, Nian Shao, and Xiaofei Li. Self-supervised audio teacher-student Transformer for both clip-level and frame-level tasks. *arXiv preprint arXiv:2306.04186*, 2023.

[Lillicrap *et al.*, 2015] Timothy Lillicrap, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, et al. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[Loshchilov and Hutter, 2016] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Ma *et al.*, 2023] Ziyang Ma, Zhisheng Zheng, Changli Tang, Yujin Wang, and Xie Chen. MT4SSL: Boosting self-supervised speech representation learning by integrating multiple targets. In *Proc. Interspeech*, 2023.

[Nagrani *et al.*, 2021] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *Proc. NeurIPS*, 2021.

[Niizumi *et al.*, 2021] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. BYOL for audio: Self-supervised learning for general-purpose audio representation. In *Proc. IJCNN*. IEEE, 2021.

[Niizumi *et al.*, 2022] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. In *HEAR: Holistic Evaluation of Audio Representations*, pages 1–24. PMLR, 2022.

[Niizumi *et al.*, 2023] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Masked modeling duo: Learning representations by encouraging both networks to model the input. In *Proc. ICASSP*. IEEE, 2023.

[Ott *et al.*, 2019] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. Fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

[Park *et al.*, 2019] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[Piczak, 2015] Karol J Piczak. ESC: Dataset for environmental sound classification. In *Proc. ACM MM*, 2015.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[Srivastava *et al.*, 2022] Sangeeta Srivastava, Yun Wang, Andros Tjandra, Anurag Kumar, Chunxi Liu, Kritika Singh, and Yatharth Saraf. Conformer-based self-supervised learning for non-speech audio tasks. In *Proc. ICASSP*. IEEE, 2022.

[Warden, 2018] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[Wu *et al.*, 2022] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *Proc. ICASSP*. IEEE, 2022.

[Yang *et al.*, 2021] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng Jeff Lai, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*, 2021.

[Zhang *et al.*, 2017] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.