BEATs: Audio Pre-Training with Acoustic Tokenizers

Sanyuan Chen [1] **Yu Wu** [2] **Chengyi Wang** [3] **Shujie Liu** [2] **Daniel Tompkins** [4] **Zhuo Chen** [4] **Wanxiang Che** [1] **Xiangzhan Yu** [1] **Furu Wei** [2]

Abstract

We introduce a self-supervised learning (SSL) framework BEATs for general audio representation pre-training, where we optimize an acoustic tokenizer and an audio SSL model by iterations. Unlike the previous audio SSL models that employ reconstruction loss for pre-training, our audio SSL model is trained with the discrete label prediction task, where the labels are generated by a semantic-rich acoustic tokenizer. We propose an iterative pipeline to jointly optimize the tokenizer and the pre-trained model, aiming to abstract high-level semantics and discard the redundant details for audio. The experimental results demonstrate our acoustic tokenizers can generate discrete labels with rich audio semantics and our audio SSL models achieve state-of-the-art (SOTA) results across various audio classification benchmarks, even outperforming previous models that use more training data and model parameters significantly. Specifically, we set a new SOTA mAP 50.6% on AudioSet-2M without using any external data, and 98.1% accuracy on ESC-50. The code and pre-trained models are available at https://aka.ms/beats.

1. Introduction

Recent years have witnessed great success in self-supervised learning (SSL) for speech and audio processing. The speech SSL models, such as Wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), BigSSL (Zhang et al., 2022), WavLM (Chen et al., 2022b), and data2vec (Baevski et al., 2022), show prominent performance across various speech processing tasks, especially in low-resource scenarios. Unlike speech, audio typically contains wide variations of environmental events, including human voices, nature sounds, musical beats, etc, which brings great challenges to general audio modeling. Existing work (Gong et al., 2022a) finds that applying the speech SSL model directly to the audio domain does not lead to satisfactory performance. Thus, it is non-trivial to study general audio SSL methods.

Until now, state-of-the-art (SOTA) audio SSL models (Xu et al., 2022; Chong et al., 2022) employ an acoustic feature reconstruction loss as the pre-training objective instead of the discrete label prediction pre-training task as in SSL models of speech (Hsu et al., 2021; Chen et al., 2022b), vision (Bao et al., 2021; Peng et al., 2022; Wang et al., 2022b) and language (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019). However, it was generally believed that the reconstruction loss only accounts for the correctness of low-level time-frequency features but neglects high-level audio semantic abstraction (Ramesh et al., 2021; Bao et al., 2021). The discrete label prediction would be a potentially better audio pre-training objective than reconstruction for the following reasons. Firstly, from the bionics aspect, humans understand audio by extracting and clustering the high-level semantics instead of focusing on the low-level time-frequency details (Patterson et al., 2007; Harb & Chen, 2007; Ma et al., 2018). By mimicking the semantics extracting and clustering through the discrete label prediction pre-training objective, the audio SSL model is expected to learn the same understanding and generalization skills as humans. Secondly, the discrete label prediction objective can improve the audio modeling efficiency by providing semantic-rich tokens as the pre-training targets and encouraging the model to discard the redundant details, resulting in a superior audio understanding capability with a lower pre-training recourse cost (Bao et al., 2021; 2022; Peng et al., 2022). Thirdly, the audio SSL pre-training with the discrete label prediction objective advances the unification of language, vision, speech, and audio pre-training, and enables the possibility of building a foundation model across modalities with a single pre-training task, i.e. discrete label prediction (Wang et al., 2022b).

Despite these advantages and great successes in various domains, the application of discrete label prediction in gen-

[1]Harbin Institute of Technology, Harbin, Heilongjiang, China [2]Microsoft Research Asia, Beijing, China [3]Nankai University, Tianjin, China [4]Microsoft Corporation, Redmond, WA, USA. Correspondence to: Sanyuan Chen <sychen@ir.hit.edu.cn>, Yu Wu <yuwu1@microsoft.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

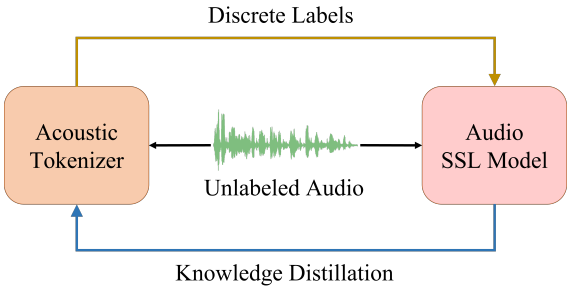


Figure 1. Iterative audio pre-training of BEATs.

eral audio processing remains challenging for two reasons. Firstly, as the audio signal is continuous and the same acoustic event might have various durations in different occasions, it is not straightforward to directly split the audio into semantically meaningful tokens as in language processing (Devlin et al., 2019). On the other hand, different from speech, the general audio signals contain excessively larger data variations, including various non-speech acoustic events and environmental sounds, where the commonly used speech tokenizer for phoneme information extraction (Hsu et al., 2021) can not be directly applied.

To tackle these challenges, in this work we propose BEATs, short for **B**idirectional **E**ncoder representation from **A**udio **T**ransformer**s**, in which an acoustic tokenizer and an audio SSL model are optimized through an iterative audio pre-training framework. The training pipeline is illustrated in Figure 1. In each iteration, we first use the acoustic tokenizer to generate the discrete labels of the unlabeled audio, and use them to optimize the audio SSL model with a mask and discrete label prediction loss. After convergence, the audio SSL model acts as a teacher to guide the acoustic tokenizer to learn audio semantics with knowledge distillation (Hinton et al., 2015). In this alternating update learning process, the acoustic tokenizer and the audio SSL model can benefit from each other. The procedure is repeated until convergence. Specifically, in the first iteration, we use a random-projection acoustic tokenizer to generate discrete labels as a cold start. In addition, we could fine-tune the audio SSL model with a little supervised data, and use the fine-tuned model as the teacher for acoustic tokenizer training. A fine-tuned model learns semantic knowledge not only from SSL but supervised learning, which can further improve the tokenizer quality. We believe the proposed pre-training framework encourages our audio SSL model to learn relevant semantic information from iterations. Our pre-training framework is also compatible with any masked audio prediction model, regardless of what backbone network is used.

We employ the vanilla ViT model (Dosovitskiy et al., 2021) as the backbone of our audio SSL models without heavy structure engineering, and apply the speed-up technique proposed in He et al. (2022). Given the discrete labels generated by the acoustic tokenizer, we mask 75% of the input sequence and let the model predict the corresponding discrete labels on mask regions. We follow Xu et al. (2022) to fine-tune the audio SSL model across various audio tasks. Experimental results show that our BEATs pre-trained models have superior performance compared with previous works across six audio and speech classification tasks. We achieve the SOTA audio understanding performance on AudioSet-2M, even outperforming the previous SOTA results that are obtained with much more model parameters (90M v.s. 304M) and training data (e.g. ImageNet) by a large margin (48.6% v.s. 47.4% for single model and 50.6% v.s. 49.6% for ensemble models). On ESC-50, our BEATs also achieved 25% relative error rate reduction over the previous SOTA performance. We further demonstrate the effectiveness of our proposed acoustic tokenizers, where the generated discrete labels are robust to random disturbances and well aligned with audio semantics.

Our contributions include the following: 1) We propose an iterative audio pre-training framework, which opens the door to audio pre-training with a discrete label prediction loss and shows better performance than with reconstruction loss. It unifies the pre-training for speech and audio, which sheds light on the foundation model building for both speech and audio. 2) We provide effective acoustic tokenizers to quantize continuous audio features into semantic-rich discrete labels, facilitating future work of audio pre-training and multi-modality pre-training. 3) We achieve SOTA results on several audio and speech understanding benchmarks. The models and codes are released to facilitate future research[1].

2. Related Work

Recently, audio pre-training has achieved great success in audio understanding tasks. The existing audio pre-training methods include supervised pre-training and self-supervised pre-training. Previous works (Gong et al., 2021a;b; Koutini et al., 2021; Nagrani et al., 2021; Chen et al., 2022a) find supervised pre-training with out-of-domain data (e.g. ImageNet (Deng et al., 2009)) can obtain significant accuracy improvement on audio understanding tasks. Some methods (Kong et al., 2020; Gong et al., 2021a; Koutini et al., 2021; Verbitskiy et al., 2022; Chen et al., 2022a; Xu et al., 2022; Elizalde et al., 2022; Wu et al., 2022; Guzhov et al., 2022) also leverage in-domain data (e.g. AudioSet (Gemmeke et al., 2017)) for supervised audio pre-training. Despite the promising classification results, these methods strongly rely on a great amount of supervised data, which is complex and expensive in practice. In comparison, the self-supervised pre-training methods only require large-scale

[1] https://aka.ms/beats

unlabeled data, which can be easily get from the Internet. The self-supervised audio pre-training methods typically learn the audio representations with the contrastive learning (Ravanelli & Bengio, 2018; Saeed et al., 2021; Fonseca et al., 2021; Al-Tahan & Mohsenzadeh, 2021; Wang & Oord, 2021) or reconstruction objective (Tagliasacchi et al., 2020; Niizumi et al., 2021; 2022; Gong et al., 2022a; Baade et al., 2022; Chong et al., 2022; Xu et al., 2022). Until now, the MAE-style reconstruction pre-training methods (Baade et al., 2022; Niizumi et al., 2022; Chong et al., 2022; Xu et al., 2022) show the best audio understanding performance on various audio classification tasks. Unlike the previous methods, we explore the self-supervised audio pre-training method with the masked discrete label prediction objective for the first time.

Various tokenizers have been proposed for learning discrete representations on audio and speech tasks. Dieleman et al. (2018) propose a hierarchical VQ-VAE based model to learn audio discrete representations for music generation tasks. HuBERT (Hsu et al., 2021) generates discrete labels with the iterative hidden state clustering method for speech SSL task, where the hidden state is extracted from the last round speech SSL model. Chiu et al. (2022) claim a random-projection tokenizer is adequate for a large speech SSL model pre-training. Our work is the first to train an acoustic tokenizer with the supervision of the last round SSL model, which is different from the previous auto-encoding and ad-hoc clustering methods.

3. BEATs

3.1. Iterative Audio Pre-training

Figure 1 shows the overall pipeline of our iterative audio pre-training framework of BEATs, where an acoustic tokenizer (Section 3.2) and an audio SSL model (Section 3.3) are optimized by iterations. In each iteration, given the unlabeled audio, we use the acoustic tokenizer to generate the discrete labels, and use them to train the audio SSL model with a mask and discrete label prediction loss. After model convergence, we use the audio SSL model as the teacher to train a new acoustic tokenizer with knowledge distillation for the next iteration of audio SSL model training.

Specifically, given an audio clip as the input, we first extract the corresponding acoustic features, split them into regular grid patches, and further flatten them to the patch sequence $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$. For the audio SSL model training, we use the acoustic tokenizer to quantize the patch sequence $\mathbf{X}$ to the patch-level discrete labels $\hat{Z} = \{\hat{z}_t\}_{t=1}^{T}$ as the masked prediction targets. For the acoustic tokenizer training, we leverage the audio SSL model to encode the patch sequence $\mathbf{X}$ and extract the output sequence $\hat{\mathbf{O}} = \{\hat{\mathbf{o}}_t\}_{t=1}^{T}$ as the knowledge distillation targets.

Note that we could leverage either a pre-trained audio SSL model or a fine-tuned audio SSL model as the teacher for acoustic tokenizer training. A fine-tuned model learns semantic knowledge not only from self-supervised pre-training but supervised fine-tuning, making it a better teacher for audio semantics distillation. With this alternating update learning process, the acoustic tokenizer benefits from the semantic-rich knowledge encoded by the audio SSL model, while the audio SSL model benefits from semantic-rich discrete labels generated by the acoustic tokenizer. The procedure is repeated until convergence, and the theoretical proof of convergence is provided in Appendix A.

3.2. Acoustic Tokenizers

The acoustic tokenizers are used to generate the discrete labels for each iteration of BEATs pre-training. In the first iteration, given the teacher model is unavailable, we employ a Random-Projection Tokenizer (Section 3.2.1) to cluster the continuous acoustic features into discrete labels as a cold start. Starting from the second iteration, we train a Self-Distilled Tokenizer (Section 3.2.2) to generate the refined discrete labels with the semantic-aware knowledge distilled from the pre-trained/fine-tuned audio SSL model obtained in the last iteration.

3.2.1. Cold Start: Random-Projection Tokenizer

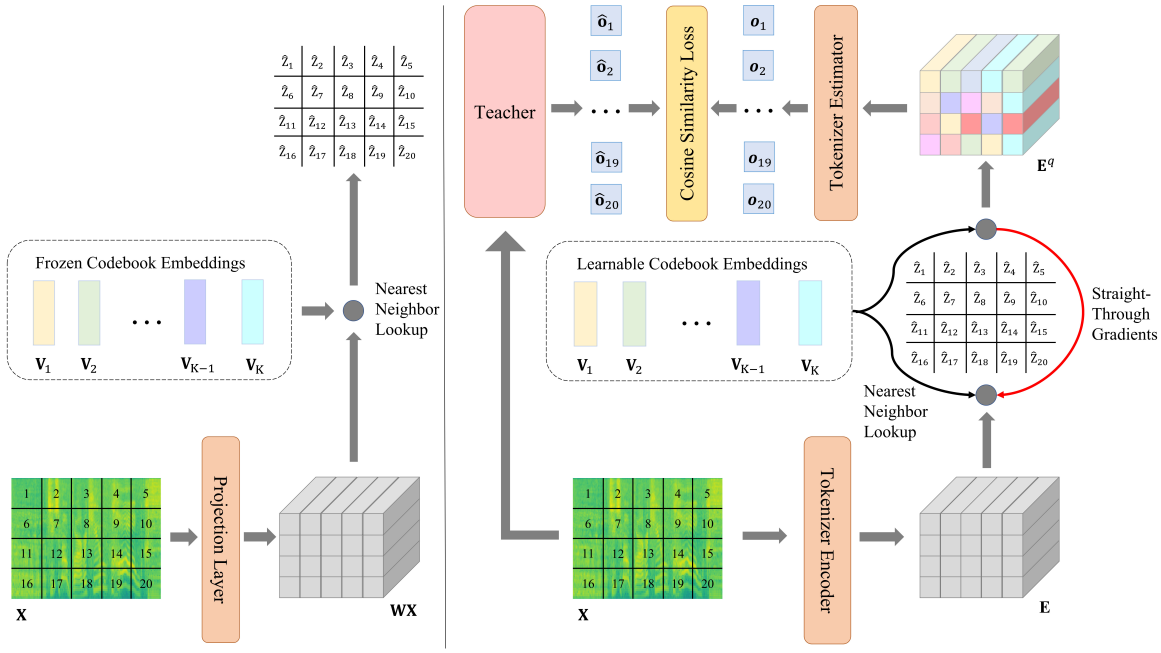
For the first iteration of BEATs pre-training, we apply the random-projection tokenizer (Chiu et al., 2022) to generate the patch-level discrete labels for each input audio.

As shown in the left part of Figure 2, the random-projection tokenizer includes a linear projection layer and a set of codebook embeddings, which are kept frozen after random initialization. Each patch of the input feature is first projected with the linear layer, then finds the nearest neighbor vector among the codebook embeddings, where the index of the nearest neighbor is defined as the discrete label.

Specifically, given the patch sequence extracted from the input audio $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$, we first project $\mathbf{x}_t$ to the vector $\mathbf{W}\mathbf{x}_t$ with a randomly initialized projection layer $\mathbf{W}$. Then we look up the nearest neighbor vector of each projected vector $\mathbf{W}\mathbf{x}_t$ from a set of random initialized vectors $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{K}$, where K is the codebook size, and define the discrete label of t-th patch as the index of the nearest neighbor vector: $\hat{z}_t = \arg\min_i ||\mathbf{v}_i - \mathbf{W}\mathbf{x}_t||_2^2$.

3.2.2. Iteration: Self-Distilled Tokenizer

From the second iteration of BEATs pre-training, we leverage the last iteration audio SSL model as the teacher, which can be either a pre-trained model or a fine-tuned model, to teach the current iteration tokenizer learning. We call it the



(a) Random-Projection Tokenizer (b) Self-Distilled Tokenizer

Figure 2. Acoustic tokenizers for discrete label generation.

self-distilled tokenizer to generate the patch-level discrete labels for each input audio.

As shown in the right part of Figure 2, the self-distilled tokenizer first uses a Transformer-based tokenizer encoder to convert the input patches to discrete labels with a set of learnable codebook embeddings. Then, a Transformer-based tokenizer estimator is trained to predict the output of a teacher model with the discrete labels and codebook embeddings as the input. With knowledge distillation as the training target, the tokenized discrete labels are optimized to contain more semantic-rich knowledge from the teacher and less redundant information of the input audio.

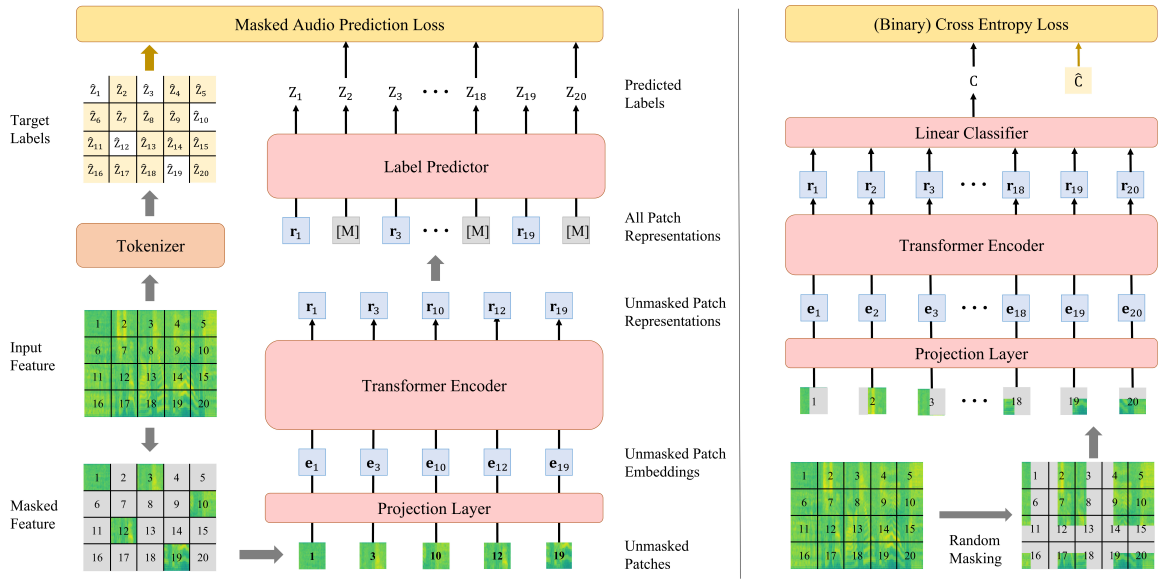
Specifically, we first feed the input patches $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$ to a 12-layer Transformer encoder and obtain the encoded vector sequence $\mathbf{E} = \{\mathbf{e}_t\}_{t=1}^T$. Then, for each encoded vector $\mathbf{e}_t$, we conduct the quantization by finding the nearest neighbor vector $\mathbf{v}_{\hat{z}_t}$ from the codebook embeddings $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^K$: $\hat{z}_t = \arg\min_i ||\ell_2(\mathbf{v}_i) - \ell_2(\mathbf{e}_t)||_2^2$, where ℓ_2 normalization is used to improve the codebook utilization (Yu et al., 2021; Peng et al., 2022). With the quantized vector sequence $\mathbf{E}^q = \{\mathbf{v}_{\hat{z}_t}\}_{t=1}^T$ as the input, we use a 3-layer Transformer estimator to predict the last layer output of the teacher model $\{\hat{\mathbf{o}}_t\}_{t=1}^T$. To deal with the non-differentiable problem of the vector quantization, following Van Den Oord et al. (2017), we apply the straight-through gradients mechanism, where the gradients are directly copied from the quantized vector sequence $\mathbf{E}^q$ to the encoded vector sequence $\mathbf{E}$ during the backward process.

The overall training objective of the self-distilled tokenizer is defined as the cosine similarity between the output sequence of the tokenizer estimator $\{\mathbf{o}_t\}_{t=1}^T$ and the output sequence of the teacher model $\{\hat{\mathbf{o}}_t\}_{t=1}^T$, along with the mean squared error between the encoded vector sequence $\mathbf{E} = \{\mathbf{e}_t\}_{t=1}^T$ and the quantized vector sequence $\mathbf{E}^q = \{\mathbf{v}_{\hat{z}_t}\}_{t=1}^T$: $\mathcal{L} = \max \sum_{\mathbf{X}\in\mathcal{D}} \sum_{t=1}^T \cos(\mathbf{o}_t, \hat{\mathbf{o}}_t) - ||sg[\ell_2(\mathbf{e}_t)] - \ell_2(\mathbf{v}_{\hat{z}_t})||_2^2 - ||\ell_2(\mathbf{e}_t) - sg[\ell_2(\mathbf{v}_{\hat{z}_t})]||_2^2$, where $\mathcal{D}$ denotes the pre-training datasets, $\cos(\cdot,\cdot)$ and $sg[\cdot]$ are the cosine similarity and the stopgradient operator, respectively. We employ the exponential moving average (Van Den Oord et al., 2017) for codebook embedding optimization for more stable tokenizer training (Peng et al., 2022). During inference, we discard the tokenizer estimator, and leverage the pre-trained tokenizer encoder and codebook embeddings to convert each input audio $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$ to patch-level discrete labels $\hat{Z} = \{\hat{z}_t\}_{t=1}^T$.

3.3. Audio SSL Model

3.3.1. Backbone

Following the previous works (Gong et al., 2021a; 2022a; Xu et al., 2022), we employ the ViT structure (Dosovitskiy et al., 2021) as the backbone network, which consists of a linear projection layer and a stack of Transformer encoder layers. Given the input audio patches $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$, we first convert them to the patch embeddings $\mathbf{E} = \{\mathbf{e}_t\}_{t=1}^T$



(a) Pre-training (b) Fine-tuning

Figure 3. Overview of audio SSL model pre-training and fine-tuning.

with a linear projection network. Then, we feed the patch embeddings to the Transformer encoder layers, and obtain the encoded patch representations $\mathbf{R} = \{\mathbf{r}_t\}_{t=1}^{T}$. The Transformer is equipped with a convolution-based relative position embedding layer at the bottom, and the gated relative position bias (Chi et al., 2022) for better position information encoding. We also employ the DeepNorm (Wang et al., 2022a) for more stable pre-training.

3.3.2. Pre-Training

We propose a Masked Audio Modeling (MAM) task for the audio SSL model pre-training, as shown in the left part of Figure 3. Different from the previous audio pre-training methods, where the model is optimized to reconstruct the input acoustic feature, our model is optimized to predict the patch-level discrete labels generated by the acoustic tokenizers (Section 3.2) with a label predictor.

Specifically, given the input patch sequence $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ and the corresponding target discrete acoustic labels $\hat{Z} = \{\hat{z}_t\}_{t=1}^{T}$, we randomly mask 75% of the input patches, where the masked positions are denoted as $\mathcal{M} = \{1, \ldots, T\}^{0.75T}$. Then, we feed the unmasked patch sequence $\mathbf{X}^U = \{\mathbf{x}_t : t \in \mathcal{M}\}_{t=1}^{T}$ to the ViT encoder, and obtain the encoded representations $\mathbf{R}^U = \{\mathbf{r}_t : t \in \mathcal{M}\}_{t=1}^{T}$. Finally, we feed the combination of the non-masked patch representations and the masked patch features $\{\mathbf{r}_t : t \in \mathcal{M}\}_{t=1}^{T} \cup \{\mathbf{0} : t \notin \mathcal{M}\}_{t=1}^{T}$ to the label predictor to predict the discrete acoustic labels $Z = \{z_t\}_{t=1}^{T}$. It should be noted that only feeding the non-masked patches into the encoder could significantly speed up the training process while providing slight improvement across downstream tasks (Xu et al., 2022). The pre-training objective of MAM is the cross entropy loss which maximizes the log-likelihood of the correct acoustic labels in the masked positions given the unmasked patch sequences: $\mathcal{L}_{\text{MAM}} = -\sum_{t \in \mathcal{M}} \log p(\hat{z}_t | \mathbf{X}^U)$.

3.3.3. Fine-Tuning

During audio SSL model fine-tuning, we discard the label predictor, and append a task-specific linear classifier upon the ViT encoder to generate the labels for the downstream classification tasks, as shown in the right part of Figure 3.

Specifically, we first random mask the input acoustic feature in the time and frequency dimension as spec-augmentation (Park et al., 2019), then split and flat it to the patch sequence $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$. Unlike pre-training, we feed the whole patch sequence $\mathbf{X}$ to the ViT encoder, and obtain the encoded representations $\mathbf{R} = \{\mathbf{r}_t\}_{t=1}^{T}$. Finally, we use a linear classifier to calculate the category probabilities as $p(C) = \text{Softmax}(\text{MeanPool}(\mathbf{W}_c\mathbf{R}))$, where Softmax, MeanPool and $\mathbf{W}_c$ denote the softmax operation, mean-pooling layer and the linear projection, respectively. We employ the cross entropy loss as the fine-tuning objective for the single label classification tasks, and the binary cross entropy loss for the multi-label classification tasks or the mixup augmentation (Zhang et al., 2017) is employed.

4. Experiment

4.1. Setup

Datasets We pre-train our BEATs tokenizers and audio SSL models on the full training set of the AudioSet dataset (Gemmeke et al., 2017), and evaluate our pre-trained audio SSL models on six downstream tasks, including three audio classification tasks (AS-2M, AS-20K (Gemmeke et al., 2017) and ESC-50 (Piczak, 2015)) and three speech classification tasks (KS1, KS2 (Warden, 2018) and ER (Busso et al., 2008)). Please see Appendix B for the detailed introduction of each downstream task.

Backbone The BEATs models have 12 Transformer encoder layers, 768-dimensional hidden states, and 8 attention heads, resulting in 90M parameters. We keep the model size similar to the previous SOTA audio pre-trained models (Xu et al., 2022; Chong et al., 2022) for a fair comparison of the pre-training methods.

Acoustic feature Following (Gong et al., 2021a; 2022a), we convert the sample rate of each raw waveform to 16,000 Hz, and extract the 128-dimensional Mel-filter bank features with a 25ms Povey window that shifts every 10 ms as the acoustic feature. The acoustic feature is normalized to the mean value of 0 and standard deviation of 0.5 following the previous works. We split each acoustic feature into the 16 $\times$ 16 patches, and further flat them to the patch sequence as the input of our BEATs tokenizers and models.

Model and tokenizer training We pre-train the BEATs models on AS-2M dataset for three iterations and denote them as $\text{BEATs}_{\text{iter1}}$, $\text{BEATs}_{\text{iter2}}$, $\text{BEATs}_{\text{iter3}}$, $\text{BEATs}_{\text{iter3+}}$.

The $\text{BEATs}_{\text{iter1}}$ is pre-trained with the discrete labels generated by a random-projection tokenizer (Section 3.2.1). Starting from the second iteration, we train a self-distilled tokenizer (Section 3.2.2) to generate the discrete labels for the pre-training of $\text{BEATs}_{\text{iter2}}$ and $\text{BEATs}_{\text{iter3}}$ with the pre-trained $\text{BEATs}_{\text{iter1}}$ and $\text{BEATs}_{\text{iter2}}$ as the teacher, respectively. Different from $\text{BEATs}_{\text{iter3}}$, the self-distilled tokenizer for $\text{BEATs}_{\text{iter3+}}$ pre-training takes the supervised fine-tuned $\text{BEATs}_{\text{iter2}}$ as the teacher model and learns to estimate the classification logits of the input audios. Compared with the other BEATs models, the $\text{BEATs}_{\text{iter3+}}$ not only make use of the downstream supervised data during fine-tuning but also in pre-training.

We pre-train all the BEATs models for 400k steps with a batch size of 5.6K seconds and a 5e-4 peak learning rate. The codebook of all the tokenizers contains 1024 embeddings with 256 dimensions. The self-distilled tokenizer with a self-supervised model as the teacher is trained for 400k steps with a batch size of 1.4K seconds and a 5e-5 peak learning rate. The self-distilled tokenizer with a supervised model as the teacher is trained for 400k steps with a batch size of 1.4K seconds and a 5e-4 peak learning rate. Each of the BEATs models is trained with 16 Tesla V100-SXM2-32GB GPUs for around 75 hours and the self-distilled tokenizer is trained with 8 Tesla V100-SXM2-32GB GPUs for around 45 hours. Please see Appendix C for the detailed hyperparameter settings.

4.2. Comparing with the SOTA Single Models

Table 1 shows the comparison of the single-model performance of our BEATs pre-trained models and the previous SOTA models. For a fair comparison with the previous self-supervised pre-training methods, we report the $\text{BEATs}_{\text{iter3+}}$ fine-tuning results on AS-2M and AS-20K with the models that are pre-trained with the same supervised dataset as fine-tuning. On the other tasks, we report the $\text{BEATs}_{\text{iter3+}}$ fine-tuning results with the model that is pre-trained with the AS-2M supervised dataset, and compare them with the previous supervised pre-training methods. Following (Xu et al., 2022; Gong et al., 2021a), we report the $\text{BEATs}_{\text{iter3+}}$ fine-tuning result on ESC-50 with additional supervised training on AS-2M.

Overall, BEATs achieve the best performance across all six audio and speech classification tasks. $\text{BEATs}_{\text{iter3+}}$ set a new SOTA single-model audio understanding performance on AS-2M and AS-20K, and outperform the previous SOTA results by a large margin (48.6 v.s. 47.4 on AS-2M, and 38.9 v.s. 37.6 on AS-20K) with much fewer model parameters (90M v.s. 304M). Notably, BEATs also significantly outperform all the previous models that use more out-of-domain or in-domain data for supervised or self-supervised pre-training. On ESC-50, BEATs successfully reduce the SOTA classification error rate from 5.9% to 4.4% without any external supervised data, and from 2.6% to 1.9% with external AS-2M supervised data.

As shown in the table, our first iteration model $\text{BEATs}_{\text{iter1}}$ which uses a random-projection tokenizer for label generation can already obtain better performance than previous works on five out of six tasks (AS-2M, ESC-50, KS1, KS2, and ER), which demonstrates the superiority of the discrete label prediction loss comparing to the reconstruction loss. Pre-trained with the refined labels generated by a self-distilled tokenizer, $\text{BEATs}_{\text{iter2}}$ can achieve further performance improvements, especially on the audio classification tasks. With SSL on AS-2M, $\text{BEATs}_{\text{iter1}}$ learns to encode the high-level audio representations with semantic-aware knowledge. Taking $\text{BEATs}_{\text{iter1}}$ as the teacher model, the self-distilled tokenizer is optimized to refine the labels with more audio-related semantics, resulting in the more powerful audio modeling ability of $\text{BEATs}_{\text{iter2}}$.

As for the third iteration of BEATs pre-training, we can find that $\text{BEATs}_{\text{iter3}}$ obtains similar performance as $\text{BEATs}_{\text{iter2}}$,

Table 1. Comparing with the SOTA single models on audio and speech classification tasks. IN, AS, and LS denote the ImageNet, AudioSet, and LibriSpeech (Panayotov et al., 2015) datasets, respectively. TA and TI denote the 128K text-audio pairs and 400M text-image pairs for CLAP (Elizalde et al., 2022) and CLIP (Radford et al., 2021) pre-training, respectively. The evaluation metrics are mAP for AS-2M/AS-20K and accuracy for ESC-50/KS1/KS2/ER. We compared the best single models from each previous work. We gray-out the models and results with additional supervised training on the external datasets. *The results reported following the SUPERB policy (wen Yang et al., 2021), where pre-trained models are kept frozen during fine-tuning.

Model	# Param	Data	Audio			Speech		
			AS-2M	AS-20K	ESC-50	KS1	KS2	ER
No Pre-Training								
PANN (Kong et al., 2020)	81M	-	43.1	27.8	83.3	-	61.8	-
ERANN (Verbitskiy et al., 2022)	55M	-	45.0	-	89.2	-	-	-
Out-of-domain Supervised Pre-Training								
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	95.5	98.1	56.0
MBT (Nagrani et al., 2021)	86M	IN-21K	44.3	31.3	-	-	-	-
PaSST (Koutini et al., 2021)	86M	IN	47.1	-	-	-	-	-
HTS-AT (Chen et al., 2022a)	31M	IN	47.1	-	-	-	98.0	-
Wav2CLIP (Wu et al., 2022)	74M	TI+AS	-	-	86.0	-	-	-
AudioCLIP (Guzhov et al., 2022)	93M	TI+AS	25.9	-	96.7	-	-	-
In-domain Supervised Pre-Training								
PANN (Kong et al., 2020)	81M	AS	-	-	94.7	-	-	-
ERANN (Verbitskiy et al., 2022)	55M	AS	-	-	96.1	-	-	-
AST (Gong et al., 2021a)	86M	IN+AS	45.9	-	95.6	-	97.9	-
PaSST (Koutini et al., 2021)	86M	IN+AS	47.1	-	96.8	-	-	-
HTS-AT (Chen et al., 2022a)	31M	IN+AS	47.1	-	97.0	-	-	-
CLAP (Elizalde et al., 2022)	190.8M	TA	-	-	96.7	-	96.8	-
Audio-MAE (Xu et al., 2022)	86M	AS	-	-	97.4	-	-	-
Self-Supervised Pre-Training								
Wav2vec (Schneider et al., 2019)	33M	LS	-	-	-	96.2	-	59.8
Wav2vec 2.0 (Baevski et al., 2020)	95M	LS	-	-	-	96.2*	-	63.4*
SS-AST (Gong et al., 2022a)	89M	AS+LS	-	31.0	88.8	96.0	98.0	59.6
MSM-MAE (Niizumi et al., 2022)	86M	AS	-	-	85.6	-	87.3	-
MaskSpec (Chong et al., 2022)	86M	AS	47.1	32.3	89.6	-	97.7	-
MAE-AST (Baade et al., 2022)	86M	AS+LS	-	30.6	90.0	95.8	97.9	59.8
Audio-MAE (Xu et al., 2022)	86M	AS	47.3	37.1	94.1	96.9	**98.3**	-
data2vec (Baevski et al., 2022)	94M	AS	-	34.5	-	-	-	-
Audio-MAE Large (Xu et al., 2022)	304M	AS	47.4	37.6	-	-	-	-
CAV-MAE (Gong et al., 2022b)	86M	AS+IN	44.9	34.2	-	-	-	-
Ours								
BEATs$_{\text{iter1}}$	90M	AS	47.9	36.0	94.0	**98.0**	**98.3**	65.9
BEATs$_{\text{iter2}}$	90M	AS	48.1	38.3	95.1	97.7	**98.3**	**66.1**
BEATs$_{\text{iter3}}$	90M	AS	48.0	38.3	**95.6**	97.7	**98.3**	64.5
BEATs$_{\text{iter3+}}$	90M	AS	**48.6**	**38.9**	98.1	98.1	98.1	65.0

indicating our self-distilled tokenizer is robust to different SSL teacher models, and our BEATs iterative pre-training procedure is capable of fast convergence in only a few iterations. Furthermore, if we use the fine-tuned BEATs$_{\text{iter2}}$ models as the teacher model, the BEATs$_{\text{iter3+}}$ can bring significant performance gains on both AS-2M and AS-20K tasks, and outperform all the previous SOTA models by a large margin. By leveraging the supervised fine-tuning data in our iterative training pipeline, both the acoustic tokenizer and the audio SSL model learn more task-specific semantic knowledge from each other, which would effectively promote BEATs$_{\text{iter3+}}$ performance on the downstream understanding tasks.

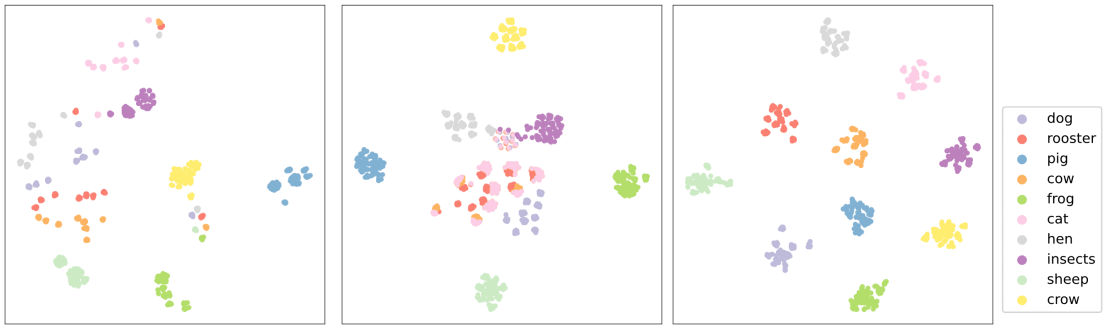
4.3. Comparing Different BEATs Tokenizers

Table 2 shows the detailed performance comparison of different BEATs tokenizers. We can find that the self-distilled tokenizer shows remarkable superiority compared with the random-projection tokenizer, especially in the task with scarce data. It is because the random-projection tokenizer with a simple feature clustering process is insufficient to provide the labels with the high-level audio semantic abstraction, while the self-distilled tokenizer is able to distill the semantic knowledge from a well pre-trained audio SSL model to the generated discrete labels.

In addition, the results show that the performance of the self-distilled tokenizer is insensitive to different self-supervised teachers (e.g. BEATs models) but sensitive to different

Table 2. Comparing different BEATs tokenizers on audio classification tasks. SSL Data and SL Data denote the training data used for self-supervised learning and supervised learning, respectively. *We use AS-2M supervised data during pre-training and AS-20K supervised data during fine-tuning. †Here, We report the ESC-50 results without additional supervised pre-training on AS-2M for a fair comparison of different tokenizers.

Model	Tokenizer Type	Tokenizer Teacher	SSL Data	SL Data	AS-2M	AS-20K	ESC-50
BEATs$_{\text{iter1}}$	Random-Projection	N/A	AS	-	47.9	36.0	94
BEATs$_{\text{iter2}}$	Self-Distilled	BEATs$_{\text{iter1}}$	AS	-	48.1	38.3	95.1
BEATs$_{\text{iter3}}$	Self-Distilled	BEATs$_{\text{iter2}}$	AS	-	48.0	38.3	95.6
BEATs$_{\text{iter3+}}$	Self-Distilled	BEATs$_{\text{iter2}}$ fine-tuned on AS-20K	AS	AS-20K	48.0	38.9	96.2
BEATs$_{\text{iter3+}}$	Self-Distilled	BEATs$_{\text{iter2}}$ fine-tuned on AS-2M	AS	AS	**48.6**	**41.8***	**97.1**†



(a) Reconstruction (b) BEATs$_{\text{iter3}}$ (c) BEATs$_{\text{iter3+}}$

Figure 4. Comparing the pre-training targets of different SSL models with audio samples from ESC-50. We visualize the acoustic features for reconstruction-based SSL models, the representations quantized by the tokenizer with a self-supervised pre-trained teacher for BEATs$_{\text{iter3}}$, and the representations quantized by the tokenizer with a supervised fine-tuned teacher for BEATs$_{\text{iter3+}}$.

supervised teachers (e.g. the fine-tuned BEATs models). The self-distilled tokenizer guided by BEATs$_{\text{iter1}}$ obtains similar performance as the tokenizer guided by BEATs$_{\text{iter2}}$. The self-distilled tokenizer guided by the AS-2M fine-tuned BEATs$_{\text{iter2}}$ model can achieve the best performance on all three audio classification tasks.

4.4. Comparing Different Pre-Training Targets via Visualization

Figure 4 shows the comparison of the pre-training targets of different SSL models with audio samples from the ESC-50 dataset. Specifically, figure 4(a) demonstrates the acoustic features which are the pre-training targets for reconstruction-based SSL models. Figure 4(b) and 4(c) illustrate the pre-training targets of BEATs$_{\text{iter3}}$ and BEATs$_{\text{iter3+}}$, which are demonstrated with the quantized representations encoded by the acoustic tokenizers with a self-supervised teacher (i.e. BEATs$_{\text{iter2}}$) and a supervised teacher (i.e. the fine-tuned BEATs$_{\text{iter2}}$), respectively. We reduced the feature dimension to 2-D by T-SNE (Van der Maaten & Hinton, 2008) for better visualization.

As the standard evaluation setting, we divide the data into a 1.6K training set and a 0.4K valid set. We use the training set for BEATs$_{\text{iter3+}}$ pre-training, and the valid set for visualization. We randomly select ten audio samples with different classification labels from the valid set, then add some random disturbance on the waveform with RIR [2] reverberations and DNS noises (Reddy et al., 2021). The points with different colors denote the audios with different classification labels, and the points with the same color denote different disturbances to the same audio.

As shown in the figures, the pre-training targets of reconstruction-based SSL models are very sensitive to random disturbances on the waveform. The acoustic feature of the same audio with different disturbances can be far apart, and the acoustic feature with different labels can be closely spaced. It indicates the pre-training targets of reconstruction-based SSL models mainly contain low-level time-frequency features and lack high-level audio semantic abstractions. In comparison, the pre-training targets of BEATs models are much more robust to the random variations. With the self-supervised pre-trained model as the teacher, the acoustic tokenizer learns to cluster the audio samples with the same semantic content and get rid of the background reverberations and noises. With the supervised fine-tuned model as the teacher, the acoustic tokenizer can successfully capture high-level semantics of audio regardless of the low-level details of redundancy, and generate semantic-rich discrete tokens for more effective BEATs model pre-training.

[2] https://www.openslr.org/28/

Table 3. Comparing with the SOTA ensemble models on AS-2M.

Model	SL Data	AS-2M
PSLA (Gong et al., 2021b)	IN+AS	47.4
AST (Gong et al., 2021a)	IN+AS	48.5
HTS-AT (Chen et al., 2022a)	IN+AS	48.7
PaSST (Koutini et al., 2021)	IN+AS	49.6
BEATs (5 models)	AS	50.4
BEATs (10 models)	AS	**50.6**

4.5. Comparing with the SOTA Ensemble Models

Table 3 shows the comparison of the ensemble-model performance of our BEATs pre-trained models and the previous SOTA models on AS-2M. We first ensemble all the five AS-2M fine-tuned BEATs models that are listed in Table 2, and denote it as BEATs (5 models). As shown in the table, without using any external supervised data (e.g. ImageNet), our BEATs (5 models) significantly outperforms the previous best ensemble models by 0.8 mAP. Then, we rerun the AS-2M fine-tuning of the five BEATs SSL models with a learning rate of 5e-5 for 100k training steps, and ensemble all the ten AS-2M fine-tuned models. The BEATs (10 models) can further improve the ensemble results and achieve 50.6 SOTA mAP performance.

5. Conclusion, Limitations, and Future Work

In this paper, we propose BEATs, an iterative audio pre-training framework for audio representation learning. Different from the previous audio SSL methods that employ reconstruction loss as the pre-training objective, we present a self-distilled tokenizer to convert continuous audio signals into discrete labels, enabling the classic mask and discrete label prediction pre-training. BEATs achieve superior performance across six audio and speech classification tasks and set new SOTA results on AudioSet-2M and ESC-50 benchmarks. Further visualization analysis illustrates the pre-training targets of BEATs models are more robust to disturbances and aligned with the semantics than reconstruction-based audio SSL models, which indicates the effectiveness of the self-distilled tokenizer and accounts for the superiority of our audio pre-training framework.

Despite these advancements, our BEATs models still have several limitations.

Computation Overhead: The iterative pre-training process employed by BEATs models results in a linear increase in computational overhead with the number of iteration steps. To mitigate this issue, we release all pre-trained tokenizers and SSL models, enabling future researchers to generate semantically-rich labels for their audio samples, fine-tune SSL models, and train new audio SSL models at a significantly reduced cost.

Data Coverage: In line with previous audio pre-training studies, our work relies solely on the AudioSet-2M dataset for pre-training, which is limited in terms of audio and speech coverage. As a future direction, we plan to broaden the pre-training data to further enhance the capabilities and effectiveness of BEATs in more general audio processing tasks. Additionally, we are interested in exploring the multi-modal domain by integrating audio with vision and language.

Model Scalability: The models pre-trained in this work are limited to 96M parameters, a considerably smaller number compared to state-of-the-art pre-trained models in speech processing. Given the significant progress that larger models have achieved in natural language processing and computer vision, we believe that scaling up our SSL model offers a promising avenue for future research.

References

Al-Tahan, H. and Mohsenzadeh, Y. Clar: Contrastive learning of auditory representations. In *International Conference on Artificial Intelligence and Statistics*, pp. 2530–2538. PMLR, 2021.

Baade, A., Peng, P., and Harwath, D. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv preprint arXiv:2203.16691*, 2022.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021.

Bao, H., Wang, W., Dong, L., and Wei, F. Vl-beit: Generative vision-language pretraining. *arXiv preprint arXiv:2206.01127*, 2022.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4): 335–359, 2008.

Chen, K., Du, X., Zhu, B., Ma, Z., Berg-Kirkpatrick, T., and Dubnov, S. 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), pp. 646–650. IEEE, 2022a.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022b.

Chi, Z., Huang, S., Dong, L., Ma, S., Zheng, B., Singhal, S., Bajaj, P., Song, X., Mao, X.-L., Huang, H.-Y., et al. Xlm-e: Cross-lingual language model pre-training via electra. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6170–6182, 2022.

Chiu, C.-C., Qin, J., Zhang, Y., Yu, J., and Wu, Y. Self-supervised learning with random-projection quantizer for speech recognition. *arXiv preprint arXiv:2202.01855*, 2022.

Chong, D., Wang, H., Zhou, P., and Zeng, Q. Masked spectrogram prediction for self-supervised audio pre-training. *arXiv preprint arXiv:2204.12768*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, 2019.

Dieleman, S., van den Oord, A., and Simonyan, K. The challenge of realistic music generation: modelling raw audio at scale. *Advances in Neural Information Processing Systems*, 31, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Elizalde, B., Deshmukh, S., Ismail, M. A., and Wang, H. Clap: Learning audio concepts from natural language supervision. *arXiv preprint arXiv:2206.04769*, 2022.

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

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Gong, Y., Chung, Y.-A., and Glass, J. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021a.

Gong, Y., Chung, Y.-A., and Glass, J. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021b.

Gong, Y., Lai, C.-I., Chung, Y.-A., and Glass, J. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10699–10709, 2022a.

Gong, Y., Rouditchenko, A., Liu, A. H., Harwath, D., Karlinsky, L., Kuehne, H., and Glass, J. Contrastive audio-visual masked autoencoder. *arXiv preprint arXiv:2210.07839*, 2022b.

Guzhov, A., Raue, F., Hees, J., and Dengel, A. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980. IEEE, 2022.

Harb, H. and Chen, L. A general audio classifier based on human perception motivated model. *Multimedia Tools and Applications*, 34:375–395, 2007.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

Koutini, K., Schlüter, J., Eghbal-zadeh, H., and Widmer, G. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ma, K. W., Wong, H. M., and Mak, C. M. A systematic review of human perceptual dimensions of sound: Meta-analysis of semantic differential method applications to indoor and outdoor sounds. *Building and Environment*, 133:123–150, 2018.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34: 14200–14213, 2021.

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

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., and Kashino, K. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. *arXiv preprint arXiv:2204.12260*, 2022.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Patterson, K., Nestor, P. J., and Rogers, T. T. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature reviews neuroscience*, 8(12):976–987, 2007.

Peng, Z., Dong, L., Bao, H., Ye, Q., and Wei, F. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. 2022.

Piczak, K. J. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Ravanelli, M. and Bengio, Y. Learning speaker representations with mutual information. *arXiv preprint arXiv:1812.00271*, 2018.

Reddy, C. K., Dubey, H., Koishida, K., Nair, A., Gopal, V., Cutler, R., Braun, S., Gamper, H., Aichner, R., and Srinivasan, S. Interspeech 2021 deep noise suppression challenge. *arXiv preprint arXiv:2101.01902*, 2021.

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

Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Verbitskiy, S., Berikov, V., and Vyshegorodtsev, V. Eranns: Efficient residual audio neural networks for audio pattern recognition. *Pattern Recognition Letters*, 161:38–44, 2022.

Wang, H., Ma, S., Dong, L., Huang, S., Zhang, D., and Wei, F. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022a.

Wang, L. and Oord, A. v. d. Multi-format contrastive learning of audio representations. *arXiv preprint arXiv:2103.06508*, 2021.

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022b.

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

wen Yang, S., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., tik Lee, K., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., and yi Lee, H. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pp. 1194–1198, 2021. doi: 10.21437/Interspeech.2021-1775.

Wu, H.-H., Seetharaman, P., Kumar, K., and Bello, J. P. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4563–4567. IEEE, 2022.

Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., Feichtenhofer, C., et al. Masked autoencoders that listen. *arXiv preprint arXiv:2207.06405*, 2022.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, Y., Park, D. S., Han, W., Qin, J., Gulati, A., Shor, J., Jansen, A., Xu, Y., Huang, Y., Wang, S., et al. Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532, 2022.

A. Convergence Analysis

In this section, we investigate the convergence properties of our proposed iterative audio pre-training procedure. From the probabilistic perspective, we can formulate the iterative pre-training of our tokenizer and SSL model with the expectation-maximization (EM) algorithm, which is used to obtain maximum likelihood estimates of the parameters in models with latent variables. By utilizing the EM algorithm, the likelihood function of the observable data is guaranteed to be non-decreasing in each iteration, ensuring the convergence of our iterative pre-training procedure. In the following, we present a mathematical analysis and formally prove the convergence of our approach.

A.1. Mathematical Formulation

Given the input audio samples $\mathbf{X}$, the tokenizer parameters $\boldsymbol{\delta}$, and the SSL model parameters $\boldsymbol{\theta}$, our framework is optimized to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta})$ with the discrete labels $\mathbf{Z}$ and SSL model representation $\mathbf{R}$ as the latent variables. We denote $\boldsymbol{\delta}^{(t)}$ and $\boldsymbol{\theta}^{(t)}$ as the tokenizer and the SSL model parameters optimized in the t-th iteration, respectively. In the $(t+1)$-th iteration, the tokenizer is trained to maximize the joint distribution $p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta})$, where $\mathbf{R}$ is sampled from the posterior distribution $p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$ estimated by the SSL model from the previous iteration. Subsequently, the SSL model is trained to maximize the joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta}^{(t+1)})$, where $\mathbf{Z}$ is sampled from the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\delta}^{(t+1)})$ estimated by the tokenizer from the current iteration.

A.2. Convergence Proof

Lemma A.1. *Given the mathematical formulation in Section A.1, the likelihood is guaranteed to be non-decreasing during each iteration of the tokenizer training. Formally,* $\forall t \in \mathbb{N}, \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$.

Proof of Lemma A.1. We derive the log-likelihood function for each iteration of the tokenizer training as follows:

$$
\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta}) &= \mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta}) \\
&= \mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \left(\log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\delta}) - \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})\right) \\
&= \mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \left(\log \frac{p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{R})} - \log \frac{p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{R})}\right) \\
&= \mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \log \frac{p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{R})} - \mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \log \frac{p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{R})} \\
&= \underbrace{\mathbb{E}_{\mathbf{R}\sim q(\mathbf{R})} \log \frac{p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{R})}}_{\text{ELBO}(q(\mathbf{R}), \boldsymbol{\theta}, \boldsymbol{\delta})} + \text{KL}(q(\mathbf{R})||p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})),
\end{aligned}
\tag{1}
$$

where a distribution $q(\mathbf{R})$ defined over the latent variables $\mathbf{R}$ is introduced, and the Kullback-Leibler (KL) Divergence between the distributions $q(\mathbf{R})$ and $p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta}))$ is denoted as $\text{KL}(q(\mathbf{R})||p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta}))$. Since the KL divergence satisfies $\text{KL}(q(\mathbf{R})||p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})) \geq 0$, with equality if, and only if $q(\mathbf{R}) = p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})$, $\text{ELBO}(q(\mathbf{R}), \boldsymbol{\theta}, \boldsymbol{\delta})$ is the evidence lower bound on the log-likelihood $\log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta})$.

In the E step of $(t+1)$-th iteration, we maximize $\text{ELBO}(q(\mathbf{R}), \boldsymbol{\theta}, \boldsymbol{\delta})$ with respect to $q(\mathbf{R})$ while keeping $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\delta} = \boldsymbol{\delta}^{(t)}$ fixed. Since $\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$ does not depend on $q(\mathbf{R})$, maximizing $\text{ELBO}(q(\mathbf{R}), \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$ is equivalent to minimizing $\text{KL}(q(\mathbf{R})||p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}))$, resulting in $q(\mathbf{R}) = p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$.

In the M step of $(t+1)$-th iteration, we maximize $\text{ELBO}(q(\mathbf{R}), \boldsymbol{\theta}, \boldsymbol{\delta})$ with respect to $\boldsymbol{\delta}$ while keeping $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ and $q(\mathbf{R}) = p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$ fixed, and obtain the optimized parameters $\boldsymbol{\delta}^{(t+1)}$ as follows.

$$
\begin{aligned}
\boldsymbol{\delta}^{(t+1)} &= \arg\max_{\boldsymbol{\delta}} \text{ELBO}(p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}), \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}) \\
&= \arg\max_{\boldsymbol{\delta}} \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})} \log \frac{p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta})}{p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})} \\
&= \arg\max_{\boldsymbol{\delta}} \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta})
\end{aligned}
\tag{2}
$$

Hence, we obtain $\forall t \in \mathbb{N}$ that

$$\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) &= \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \Big(\log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) - \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})\Big) \\
&= \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) - \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) - \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}) \\
&\geq \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}, \mathbf{R}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}) - \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}) \\
&= \mathbb{E}_{\mathbf{R}\sim p(\mathbf{R}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t)})} \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)}) \\
&= \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})
\end{aligned} \tag{3}$$

The proof of Lemma A.1 is thus complete. □

Lemma A.2. *Given the mathematical formulation in Section A.1, the likelihood is guaranteed to be non-decreasing during each iteration of the SSL model training. Formally,* $\forall t \in \mathbb{N}, \log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) \geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})$.

Proof of Lemma A.2. We derive the log-likelihood function for each iteration of the SSL model training as follows:

$$\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta}) &= \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta}) \\
&= \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \left(\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta}) - \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})\right) \\
&= \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \left(\log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{Z})} - \log \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{Z})}\right) \\
&= \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{Z})} - \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \log \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{Z})} \\
&= \underbrace{\mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta})}{q(\mathbf{Z})}}_{\text{ELBO}(q(\mathbf{Z}),\boldsymbol{\theta},\boldsymbol{\delta})} + \text{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})),
\end{aligned} \tag{4}$$

where a distribution $q(\mathbf{Z})$ defined over the latent variables $\mathbf{R}$is introduced. Since the KL divergence satisfies $\text{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})) \geq 0$, with equality if, and only if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\delta})$, $\text{ELBO}(q(\mathbf{Z}), \boldsymbol{\theta}, \boldsymbol{\delta})$ is the evidence lower bound on the log-likelihood $\log p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\delta})$.

In the E step of $(t+1)$-th iteration, we maximize $\text{ELBO}(q(\mathbf{Z}), \boldsymbol{\theta}, \boldsymbol{\delta})$ with respect to $q(\mathbf{Z})$ while keeping $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\delta} = \boldsymbol{\delta}^{(t+1)}$ fixed. Since $\log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})$ does not depend on $q(\mathbf{Z})$, maximizing $\text{ELBO}(q(\mathbf{Z}), \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})$ is equivalent to minimizing $\text{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}))$, resulting in $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})$.

In the M step of $(t+1)$-th iteration, we maximize $\text{ELBO}(q(\mathbf{Z}), \boldsymbol{\theta}, \boldsymbol{\delta})$ with respect to $\boldsymbol{\theta}$ while keeping $\boldsymbol{\delta} = \boldsymbol{\delta}^{(t+1)}$ and $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})$ fixed, and obtain the optimized parameters $\boldsymbol{\theta}^{(t+1)}$ as follows.

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \arg\max_{\boldsymbol{\theta}} \text{ELBO}(p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}), \boldsymbol{\theta}, \boldsymbol{\delta}^{(t+1)}) \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta}^{(t+1)})}{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})} \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\delta}^{(t+1)})
\end{aligned} \tag{5}$$

Hence, we obtain $\forall t \in \mathbb{N}$ that

$$
\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) &= \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \Big(\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) - \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)})\Big) \\
&= \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) - \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) - \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&\geq \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) - \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{R}|\mathbf{X}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)},\boldsymbol{\delta}^{(t+1)})} \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&= \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)})
\end{aligned}
\tag{6}
$$

The proof of Lemma A.1 is thus complete. □

Theorem A.3. *Given the mathematical formulation in Section A.1, the likelihood is guaranteed to be non-decreasing with each iteration of the iterative audio pre-training procedure. Formally,* $\forall t \in \mathbb{N}, \log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) \geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})$.

Proof of Theorem A.3. By applying Lemma A.1 and Lemma A.2, we obtain:

$$
\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}) &\geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t+1)}) \\
&\geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}, \boldsymbol{\delta}^{(t)})
\end{aligned}
\tag{7}
$$

The proof of Theorem A.3 is thus complete. □

In summary, the convergence analysis demonstrates that our iterative pre-training of the tokenizer and SSL model ensures a non-decreasing likelihood of the observable data in each iteration. Although convergence to the global maximum of the likelihood is not guaranteed, the non-decreasing property ensures the convergence of our iterative pre-training procedure to a local maximum or a saddle point. This provides a theoretical foundation for the effectiveness and stability of our proposed method.

B. Datasets

We pre-train and evaluate our BEATs models with five datasets as follows. Specifically, we pre-train the acoustic tokenizers and audio SSL models on the full training set of the AudioSet dataset, and evaluate our pre-trained audio SSL models on six downstream tasks, including three audio classification tasks (AS-2M, AS-20K and ESC-50) and three speech classification tasks (KS1, KS2 and ER).

AudioSet (AS-2M and AS-20K) (Gemmeke et al., 2017) is a large-scale audio classification dataset. It contains over 2 million 10-second YouTube clips annotated with 527 audio event classes, where each clip could be annotated with multiple audio event classes. It is officially subdivided into three partitions, including a class-wise balanced set (22,176 clips), a class-wise unbalanced set (2,042,985 clips), and an eval set (20,383 clips). Due to the constant change in YouTube video availability (e.g., videos being removed or taken down), we downloaded and processed 20,666, 1,919,153, and 18,987 clips for the balanced, unbalanced, and eval sets, respectively, which is consistent with the previous works (Baade et al., 2022).

Following the previous works, we use the combination of the 21K balanced and the 1.9M unbalanced training audios for fine-tuning in the AS-2M task, and only the 21K balanced training audios for fine-tuning in the AS-20K task. We evaluate our models on the 19K eval set with the mean average precision (mAP) evaluation metric.

Environmental Sound Classification (ESC-50) (Piczak, 2015) is an audio classification dataset that contains 2,000 5-second environmental sound recordings annotated with 50 classes. Each sound recording is only annotated with one class. We follow the 5-fold cross-validation evaluation setting as the previous works and report the classification accuracy as the evaluation metric.

Speech Commands V2 (KS2) (Warden, 2018) is a keyword spotting dataset that contains 105,829 1-second spoken word clips annotated with 35 common word classes. It is officially subdivided into the training, validation, and testing set that contains 84,843, 9,981, and 11,005 audio clips respectively. We report classification accuracy as the evaluation metric.

Speech Commands V1 (KS1) (Warden, 2018) task uses the same dataset as KS2, but only contains 10 classes of keywords, 1 silence class, and 1 unknown class that includes all the other 20 common speech commands. We use the standard data and split provided in SUPERB benchmark (wen Yang et al., 2021) to report classification accuracy for a fair comparison with the previous works.

IEMOCAP (ER) (Busso et al., 2008) is an emotion recognition dataset that contains about 12 hours of emotional speech clips annotated with four classes. we use the 5-fold cross-validation evaluation setting as SUPERB benchmark (wen Yang et al., 2021) and report classification accuracy as the evaluation metric.

C. Hyperparamter Settings

Table 4 shows the detailed hyperparameters that are used for BEATs acoustic tokenizer training, audio SSL model pre-training and fine-tuning, which are adapted from the previous works (Xu et al., 2022; Chen et al., 2022b; Peng et al., 2022).

Hyperparameters	Tokenizer Training		Model Pre-Training	Model Fine-Tuning					
	SSL Teacher	SL Teacher	AS-2M	AS-2M	AS-20K	ESC	KS1	KS2	ER
Optimizer	AdamW (Loshchilov & Hutter, 2017)								
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.98$								
Weight decay	0.01								
Learning Rate Schedule	Linear Decay			Cosine					
Steps	400K			50K			80K		
Warmup epochs	32K			5K			8K		
GPU	8		16	16			4		
Batch size (s)	1.4K		5.6K	6.4K	800	300	100		300
Layer-wise learning rate decay	1.0		1.0	0.6	0.3	0.2	0.3		1.0
Peak learning rate	5e-5	5e-4	5e-4	1e-4	3e-5		1e-4		3e-5
Weighted Sampling	✗			✓	✗		✓ *	✗	
Dropout (Srivastava et al., 2014)	0.1			0.0					
Layer Dropout	0.0			0.1					
Roll Augmentation	✗			✓			✗		✓
SpecAug (Park et al., 2019)	N/A			0.3		0.2	0.3		0.15
Mixup (Zhang et al., 2017)	N/A		0.0	0.8		0.0	0.8		0.0
Multilabel	N/A		✗	✓		✗	✗		✗
Loss Function	CosineSimilarity		CE	BCE		CE	BCE		CE
Dataset Mean for Normalization	15.41663					11.72215	11.43905	11.41045	12.0889
Dataset Std for Normalization	6.55582					10.60431	5.64913	5.67857	4.29147

Table 4. Hyperparameters of BEATs acoustic tokenizer training, audio SSL model pre-training and fine-tuning. CE and BCE denote the cross entropy loss and binary cross entropy loss, respectively. *We balance each class to 50% of the size of the unknown class for each training epoch.