Interspeech 2022
18-22 September 2022, Incheon, Korea

ATST: Audio Representation Learning with Teacher-Student Transformer

*Xian Li, Xiaofei Li**

[1]Westlake Institute for Advanced Study & [2]Westlake University , Hangzhou, China
lixian@westlake.edu.cn, lixiaofei@westlake.edu.cn

Abstract

Self-supervised learning (SSL) learns knowledge from a large amount of unlabeled data, and then transfers the knowledge to a specific problem with a limited number of labeled data. SSL has achieved promising results in various domains. This work addresses the problem of segment-level general audio SSL, and proposes a new transformer-based teacher-student SSL model, named ATST. A transformer encoder is developed on a recently emerged teacher-student baseline scheme, which largely improves the modeling capability of pre-training. In addition, a new strategy for positive pair creation is designed to fully leverage the capability of transformer. Extensive experiments have been conducted, and the proposed model achieves the new state-of-the-art results on almost all of the downstream tasks.

Index Terms: Audio pretraining, Self-supervised learning, Teacher-student model, Transformer

1. Introduction

Recently, learning audio representations with self-supervised learning (SSL) has been widely studied [1][2][3][4][5]. Among them, the contrastive learning methods [3][4][5] maximize the classification similarity of two augmented views of the same audio clip (called a positive pair), having shown a great promise for learning good representation. The above idea often confronts the issue of model collapse, e.g. the model can find an easy solution to output a constant value for any inputs. COLA [3] and [4] overcome this issue by distinguishing positive audio samples from a batch of negative audio samples. Considering the fact that for audio data, negative samples are possibly similar to positive samples in some scenarios, BYOL-A [5] proposed to discard the negative samples, and use a teacher-student scheme to overcome the issue of model collapse. The teacher and student networks process two different views of an audio clip, and the student network is trained to predict a representation being identical to the prediction of the teacher model. The teacher network is updated by taking an exponential moving average (EMA) of the student network. Another technical line for audio SSL follows the spirit of Bert [6], performing a predictive task for the masked frames, e.g. wav2vec2 [7] and Hubert [8].

Transformer network has shown powerful abilities in learning long-term dependencies, and has been used for speech SSL in several works, e.g. MockingJay [9], wav2vec2 [7], Hubert [8], Tera [10]. As for general audio SSL, people usually use convolution neural network (CNN), e.g. in COLA, BYOL-A, etc. To the best of our knowledge, SSAST [11] and [12] are the only two very recent works that use transformer for general audio SSL. They both follow the line of wav2vec2 [7].

According to the grain size of the representation at the pre-training stage, all the above methods can be categorized into two types: segment-level method and frame-level method. The segment-level methods, e.g. COLA [3] and BYOL-A [5], extract a fixed-length segment embedding from an input audio segment. On the other hand, the frame-level methods, e.g. SSAST [11] and [12], extract an individual embedding for all frames. Learning a segment embedding is suitable for a variety of segment-level audio tasks, e.g. sound event classification, music instrument classification, speaker identification, etc. COLA [3] and BYOL-A [5] have been proven very effective for segment-level general audio SSL, and have achieved the state-of-the-art performance. Although SSAST [11] and [12] are pre-trained by a frame-level criterion, the downstream tasks that they have applied to are all segment-level tasks, where average pooling is applied to obtain the segment embedding.

This work focuses on the problem of segment-level general audio SSL, and proposes a new transformer-based teacher-student SSL model, named ATST [1]. Main contributions of this work include: i) adopting transformer encoder into the baseline teacher-student scheme of BYOL-A [5], which shows a clear superiority over the CNN encoder of BYOL-A, especially for learning the long-term semantic information of speech; ii) proposing a new view creation strategy. BYOL-A uses one short segment to create two views (one positive pair). Instead, we propose to use two different long segments, which is more fit for transformer, as the network needs to learn longer temporal dependencies and to match a more distinct positive pair created by two segments. The length of segments is carefully studied to control the distinction and overlap of the two segments, which is especially important for rationalizing the difficulty of matching positive pairs. Experiments have been conducted using the large-scale Audioset [13] dataset for pre-training. Downstream tasks cover all the three types of audio signals, namely audio event, speech, and music. Ablation experiments show the effectiveness of each of the proposed modules. The proposed model as a whole achieves the new state-of-the-art results on almost all of the downstream tasks, and surpasses other methods by a large margin on some of the downstream tasks. For example, the accuracy of speaker identification is 72% versus 40.1% without finetuning, and 94.3% versus 80.8% after finetuning.

2. The Proposed Method

2.1. Baseline Teacher-Student Scheme

In this work, we adopt the teacher-student scheme as our baseline framework, which was first proposed by Bootstrap you own latent (BYOL) [14] for image pre-training, and adopted by BYOL-A [5] for audio pre-training. Given one augmented view of an audio clip, the student network is trained to predict a data representation being identical to the teacher network's prediction on one another augmented view of the same audio clip. During training, the teacher network is updated by taking the EMA of student network. Specifically, the student network, de-

* corresponding author

[1]code: https://github.com/Audio-WestlakeU/audio_ssl

10.21437/Interspeech.2022-10126

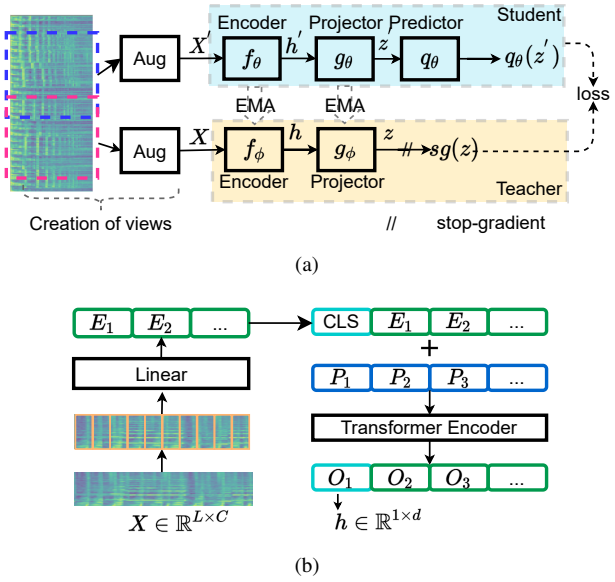

Figure 1: *(a) Overview of the proposed method. "Aug" denotes augmentation (b) Transformer encoder.*

fined by a set of weights θ, contains an encoder f_θ, a projector g_θ and a predictor q_θ, while the teacher network, defined by a set of weights ϕ, contains only an encoder f_ϕ and a projector g_ϕ. The encoders extract a representation from the augmented views. It has been shown that the additional predictor in the student network combined with the stop-gradient operation introduced by using EMA teacher network is the key factor that prevents the model from collapsing [15]. During training, ϕ is updated by the EMA of θ as: $\phi \leftarrow m\phi + (1-m)\theta$, where m is a decay rate. θ is updated as follows. Let $(\boldsymbol{X}, \boldsymbol{X}')$ be a pair of positive views created from an audio clip. $\boldsymbol{X}$ is fed into the teacher network to obtain $\boldsymbol{h} = f_\phi(\boldsymbol{X})$ and $\boldsymbol{z} = g_\phi(\boldsymbol{h})$. $\boldsymbol{X}'$ is fed into the student network to obtain $\boldsymbol{h}' = f_\theta(\boldsymbol{X}')$, $\boldsymbol{z}' = g_\theta(\boldsymbol{h}')$ and $q_\theta(\boldsymbol{z}')$. $\boldsymbol{z}$ and $q_\theta(\boldsymbol{z}')$ are then L2-norm normalized, and the mean square error between them is calculated as L_θ. A symmetric loss L'_θ is also calculated by feeding $\boldsymbol{X}$ to the student network and $\boldsymbol{X}'$ to the teacher network. During training, θ is updated by minimizing $L_\theta^{total} = L_\theta + L'_\theta$.

In BYOL-A, encoder is a CNN, projectors and predictors are multi-layer perceptrons (MLPs) that consist in a linear layer (with output dimension of 4096) followed by batch normalization, rectified linear units (RELU), and a final linear layer (with output dimension of 256).

2.2. The Proposed Method

Overview of the proposed method is depicted in Fig. 1 (a). The major differences between the proposed method and BYOL-A are two folds. The proposed method uses a transformer as encoder to leverage its powerful abilities on modeling long-term dependencies, and uses a new view (positive pair) creation strategy specifically being fit for the transformer encoder.

2.2.1. Creation of Views

BYOL-A [5] randomly crops a single 1-second segment from the input audio and then creates two views by applying different data augments to this single segment. It is considered in BYOL-A [5] that different segments may be too different to be identified as a positive pair. [4] uses two segments to create positive views, however, it uses negative views to mitigate the problem caused by using two segments.

Our view creation strategy is shown in Fig. 1 (a). The time domain input audio clip is first transformed to mel-spectrogram. We randomly crop two different segments from the mel-spectrogram. Then, two types of data augmentation are applied to each of the segments, including Mixup [5] and Random Resized Crop (RRC) [5], creating two views of the input audio clip, i.e. $(\boldsymbol{X}, \boldsymbol{X}')$. Note that this work does not use negative samples. In order to take full advantage of the transformer's ability in modeling long-term dependencies, the proposed method intends to use longer segments, e.g. 6 seconds in experiments. The proposed method separately creates two views from two different segments for the purpose of increasing the difficulty of identifying the two views as a positive pair, thus leading the model to learn more generalized representations. On the other hand, the two segments cannot be too far away from each other, otherwise the similarity between them is completely lost. This is guaranteed by properly setting the segment length to make the two segments have a certain portion of overlap. Overall, the proposed strategy does not lose the rationality of identifying two segments as a positive pair due to the overlap constraint, and meanwhile increases the task difficulty and thus the model capability by using two segments.

2.2.2. Transformer Encoder

The encoding procedure is illustrated in Fig. 1 (b). The augmented mel-spectrogram $\boldsymbol{X} \in \mathbb{R}^{L \times C}$, where L and C denote frames and channels respectively, is fed into a transformer encoder [16] to obtain a fixed-length segment embedding $\boldsymbol{h} \in \mathbb{R}^{1 \times d}$, where d denotes the dimension of embedding.

Four consecutive frames of $\boldsymbol{X}$ are first stacked to reduce the temporal resolution and sequence length. The stacked frames are fed to a linear projection layer (with output dimension of d) to obtain a new embedding sequence $\boldsymbol{E} \in \mathbb{R}^{\frac{L}{4} \times d}$ as the input sequence of transformer encoder. Besides this input sequence, we use an extra trainable class token $\boldsymbol{CLS} \in \mathbb{R}^{1 \times d}$ to represent the entire segment, which is inserted to the beginning of the input sequence. This kind of segment class token is widely used for sentence embedding in neural language processing [6], global image embedding [17], as well as audio segment embedding [18]. A trainable absolute lookup table positional embedding $\boldsymbol{P} \in \mathbb{R}^{(\frac{L}{4}+1) \times d}$ is then added to the input sequence. Eventually, we use a standard transformer encoder [16] to process the input embedding sequence, obtaining an output embedding sequence of $\boldsymbol{O} \in \mathbb{R}^{(\frac{L}{4}+1) \times d}$. In the output sequence, the segment class token, i.e. $\boldsymbol{O_1}$, aggregates information from the embedding sequence at each block of the transformer, based on the self-attention mechanism. Therefore, $\boldsymbol{O_1}$ is taken as the final segment embedding, and is denoted as $\boldsymbol{h}$ or $\boldsymbol{h}'$ in the teacher-student pre-training scheme.

As for downstream tasks, the pre-trained transformer encoder of teacher network is used as the feature extractor, while the projector is removed.

3. Experiments

We evaluate the performance of our model under the protocol of linear evaluation or finetuning. In linear evaluation, the pre-trained encoder is frozen as a feature extractor, on top of which a linear classifier is trained. Whereas in finetuning, the pre-trained encoder and linear classifier are finetuned together.

Table 1: *Linear evaluation results of **Small** model w.r.t. different view creation strategies. "Average" is taken over the last four tasks.*

Method	Segments	length of segment (s)	AS-20K mAP	SPCV2 Acc (%)	VOX1 Acc (%)	NSYNTH Acc (%)	US8K Acc (%)	Average Acc (%)
BYOL-A [5]	single	1	-	92.2	40.1	74.1	79.1	71.4
Small (Ours)	single	1	0.210	94.3	52.3	73.8	79.3	74.9
	two	1	0.191	91.3	50.0	74.3	76.6	73.1
	single	6	0.257	**94.0**	57.3	73.8	80.9	76.5
	two	6	**0.279**	93.6	**61.9**	**75.3**	**82.0**	**78.2**

3.1. Implementation Details of Pre-training

We use Audioset [13] for pre-training. The full Audioset (**AS-2M**) contains 200 million audio clips with a length of 10 seconds captured from Youtube Videos. Using the full Audioset, a **Base** model is trained, which contains 12 blocks, and 12 heads for each block. The dimension and inner dimension are 768 and 3072 respectively. Besides, using a subset with 200 thousand audio clips (**AS-200K**) randomly sampled from **AS-2M**, we also trained a **Small** model, which contains 12 blocks, and 6 heads for each block. The dimension and inner dimension are 384 and 1536 respectively.

Audio is re-sampled to 16 kHz. Audio clips are transformed to the mel-spectrogram domain, with a Hamming window, a window length of 25 ms, a hop size of 10 ms, and 64 frequency bins ranging from 60 Hz to 7800 Hz. The mel-spectrogram feature is min-max normalized, where the minimum and maximum values are calculated globally on the pre-training dataset. We intentionally set the length of two segments (for creating two views) to 6 seconds, which will leads to a segment overlap of at least 1 second, considering that the length of audio clip is 10 seconds. The two randomly sampled segments are augmented by Mixup and RRC with the same configurations used in BYOL-A [5].

We pre-train our models with the ADAMW optimizer [19]. The learning rate lr is warmed up for 10 epochs, and then annealed to 1e-6 at cosine rate. Following DINO [17], the weight decay of transformer is increased from 0.04 to 0.4 at cosine rate. The EMA decay rate m increases from an initial value m_0 to 1 at cosine rate. Batch size is set to 1536. The **Base** model is trained using **AS-2M** for 200 epochs, with lr being 2e-4, and m_0 being 0.9995. The **Small** model is trained using **AS-200K** for 300 epochs, with lr being 5e-4, and m_0 being 0.99.

3.2. Downstream Tasks

Evaluations are carried out on a variety of downstream tasks, which cover all the three types of audio signals, namely audio event, speech and music. Datasets and downstream tasks are described as follows.

- **AS-20K** for multi-label sound event classification. We use the balanced subset of Audioset-2M, with 527 audio classes. It contains 20,886 audio clips for training. For test, we use the evaluation set of Audioset, with 18,886 audio clips.
- **US8K** for single-label audio scene classification. We use the Urbansound8k dataset [20] to classify audio clips (less than 4 seconds) into 10 classes. It contains 8,732 audio clips and has ten folds for cross-validation.
- **SPCV2** for spoken command recognition. We use Speech Command V2 [21] to recognize 35 spoken commands for one second of audio. It contains 84,843, 9,981 and 11,005 audio clips for training, validation and evaluation, respectively.
- **VOX1** for speaker identification. We use the Voxceleb1 dataset [22], with 1,251 speakers. It contains 13,8361, 6,904 and 8,251 for training, validation and evaluation, respectively.
- **NSYNTH** for music instrument classification. We use the NSYNTH dataset [23], to recognize 11 instrument family classes from 4-seconds audio clips.

The mel-spectrogram feature is computed in the same way as for the pre-training data. For linear evaluation, from the pre-trained encoder, segment embedding is obtained by concatenating the class token $\boldsymbol{O_1}$ and the average of the embedding sequence of all blocks. For finetuning, segment embedding is obtained by concatenating the class token and the average of the embedding sequence of the last block. Audio clips that are longer than 12 seconds are centrally cropped with a maximum length of 12 seconds, and then split into 6-second long chunks without overlap. The chunks are independently processed by the pre-trained encoder, and their outputs are averaged to obtain the final segment embedding. Audio clips that are shorter than 6 seconds are directly processed by the pre-trained encoder to obtain the segment embedding.

For linear evaluation, we train the linear classifier for 100 epochs with the SGD optimizer. The learning rate is annealed to 1e-6 at cosine rate during training. The optimal initial learning rate is searched for each task separately. Batch size is set to 1024. Augmentation is not used.

For finetuning, we finetune all models with the SGD optimizer. The learning rate lr is warmed up for 5 epochs, and then annealed to 1e-6 at cosine rate. The optimal lr is searched for each task separately. Batch size is set to 512. We trained **SPCV2** and **VOX1** for 50 epochs, and **AS-20K** for 200 epochs. For **SPCV2** and **AS-20K**, we use Mixup [24] and RRC for data augmentation. As for supervised downstream tasks, Mixup mixes both audio clips and labels. For **VOX1**, data augmentation is not applied.

Classification accuracy (Acc) is taken as the performance metric for single-label tasks, including audio scene classification, spoken command recognition, speaker identification and music instrument classification, and mean average precision (mAP) for the task of multi-label sound event classification. For **US8K**, we conduct 10-fold cross-validation, and report the average accuracy of the 10 folds.

3.3. Ablation study

We separately evaluate the effectiveness of transformer encoder and the proposed view creation strategy. Ablation experiments are conducted using the **Small** model with the linear evaluation protocol, due to their low computational complexities. Table 1 shows the results. The result of BYOL-A is also given, which uses a CNN encoder and a single 1-second segment. Our models use a single or two segments, with a length of 1 second or

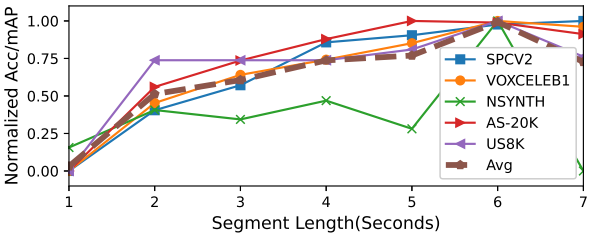

Figure 2: *Normalized Acc/mAP as a function of segment length, Acc/mAP of each task is normalized into the range of (0,1). "Avg" denotes averaging over normalized score of all tasks.*

6 seconds. For a fair comparison, when the segment length is set to 1 second, we split audio clips into 1-second chunks for downstream tasks.

Transformer Encoder: With the same view creation strategy, i.e. creating two views from a 1-second segment, our model (line 2 in Table 1) outperforms BYOL-A, especially for the two speech tasks (**SPCV2** and **VOX1**). Speech involves more long-term semantic information, and transformer is more suitable than CNN for learning these long-term dependencies.

View Creation Strategy: As shown in Table 1, when the segment length is set to 1 second, using one single segment is better than using two segments. This phenomenon is consistent with the claim made in BYOL-A [5] that the two segments may be too different to be identified as a positive pair. However, two views created from a single segment may share too much semantic content, thus leading our model to find an easy solution. When the segment length is increased to 6 seconds, the performance measures of **AS-20K**, **VOX1** and **US8K** are systematically increased, no matter whether using one or two segments. This is partially due to the capability of learning long-term dependencies of the transformer encoder. In addition, for the 6-seconds case, using two segments exhibits superior performance over using one segment. The possible reasons are: the two segments can be rationally identified as a positive pair as they share a small portion of overlap, and meanwhile they are different enough to increases the task difficulty and thus leads the model to learn a more generalized representation.

Fig. 2 shows the normalized performance of each task as a function of segment length, where two segments are used. We can see that the performance measures increase along with the increasing of segment length until 6 seconds. This further verifies our new findings: i) when transformer encoder is used, increasing the segment length helps to learn more information; ii) when two segments are used, the segment length should be set to make the segments share a proper amount of overlap, and have a proper difficulty for matching them as a positive pair.

3.4. Linear Evaluation Results

Table 2 shows the linear evaluation results. For fair comparison, we compare with other methods that also use Audioset for pre-training, including TRILL [25], COLA [3] and BYOL-A [5]. The performance scores are directly quoted from their original papers. It can be seen that our **Small** model outperforms other methods on all tasks, even though it only uses 1/10 data of Audioset-2M, while BYOL-A and COLA use the full Audioset-2M dataset. In particular, on the speaker identification task, our **Small** model obtains an accuracy of 61.9%, compared to the 40.1% accuracy of BYOL-A. By increasing the network size

Table 2: *Linear evaluation results.*

Method	AS-20K mAP	SPCV2 Acc (%)	VOX1 Acc (%)	NSYNTH Acc (%)	US8K Acc (%)
TRILL [25]	-	-	17.9	-	-
COLA [3]	-	62.4	29.9	63.4	-
BYOL-A [5]	-	92.2	40.1	74.1	79.1
Small (ours)	0.279	93.6	61.9	75.3	82.0
Base (ours)	**0.338**	**95.1**	**72.0**	**75.6**	**84.1**

Table 3: *Finetuning results.*

Method	# Params	AS-20K mAP	SPCV2 Acc (%)	VOX1 Acc (%)
COLA [3]		-	95.5	37.7
Small-SSAST [11]	23M	0.308	97.7	60.9
SSAST-PATCH [11]	89M	0.310	98.0	64.2
SSAST-FRAME [11]	89M	0.292	**98.1**	80.8
Conformer [12]	88M	0.276	-	-
Small (ours)	22M	0.315	97.6	88.3
Base (ours)	86M	**0.374**	98.0	**94.3**

and the amount of training data, the performance can be further systematically increased by our **Base** model. The superiority of the proposed model comes from the use of transformer encoder and the proposed view creation strategy, which both are critical modules for the success of contrastive learning based pretraining technique.

3.5. Fine-tuning Results

To evaluate to what extent our models can further achieve, finetuning experiments are conducted on the tasks of multi-label audio event classification (**AS-20K**), Spoken command recognition (**SPCV2**) and speaker identification (**VOX1**). We compare with those methods that also report the finetuning results, including COLA [3], SSAST [12] and Conformer [12]. Table 3 shows the results. Compared to the best results achieved by other methods, our **Base** model performs better on **AS-20K** (0.374 versus 0.310) and **VOX1** (94.3% versus 80.8%) by a large margin, and perform comparably on **SPCV2**. Remarkably, our **Small** model performs even better than SSAST and Conformer on **AS-20K** and **VOX1**, by using a much smaller network (22 M versus about 88 M). It is worth mentioning that both SSAST and Conformer use a transformer encoder as the proposed model, and use a wav2vec2-style pre-training scheme. Better results achieved by the proposed model may indicate that the teacher-student scheme is superior to the wav2vec2-style scheme for (segment-level) general audio pre-training.

4. Conclusions

In this work, we propose a general audio pre-training method with a transformer-based teacher-student scheme, named ATST. A new view creation strategy is also proposed to fully leverage the capability of transformer. We evaluate the learned representation on diverse downstream tasks. Experiments show that the proposed view creation strategy is able to improve pre-training by properly increasing the difficulty of positive pair matching. Overall, the proposed model achieves the new state-of-the-art results on almost all of the tasks. We hope our work can facilitate the progress of general audio representation learning.

5. References

[1] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv:1807.03748 [cs, stat]*, 2019.

[2] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry, and D. Roblek, "Pre-Training Audio Representations With Self-Supervision," *IEEE Signal Processing Letters*, vol. 27, pp. 600–604, 2020.

[3] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive Learning of General-Purpose Audio Representations," *arXiv:2010.10915 [cs, eess]*, 2020.

[4] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised Contrastive Learning of Sound Event Representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 371–375.

[5] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation," *arXiv:2103.06695 [cs, eess]*, 2021.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, 2019.

[7] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," *arXiv:2006.11477 [cs, eess]*, 2020.

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *arXiv:2106.07447 [cs, eess]*, 2021.

[9] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6419–6423, 2020.

[10] A. T. Liu, S.-W. Li, and H.-y. Lee, "TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[11] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-Supervised Audio Spectrogram Transformer," *arXiv:2110.09784 [cs, eess]*, 2022.

[12] S. Srivastava, Y. Wang, A. Tjandra, A. Kumar, C. Liu, K. Singh, and Y. Saraf, "Conformer-Based Self-Supervised Learning for Non-Speech Audio Tasks," *arXiv:2110.07313 [cs, eess]*, 2022.

[13] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[14] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised Learning," *arXiv:2006.07733 [cs, stat]*, 2020.

[15] X. Chen and K. He, "Exploring Simple Siamese Representation Learning," *arXiv:2011.10566 [cs]*, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, 2017.

[17] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in *arXiv:2104.14294 [cs]*, 2021.

[18] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," *arXiv:2104.01778 [cs]*, 2021.

[19] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv:1711.05101 [cs]*, 2017.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

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

[22] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[23] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

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

[25] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. d. C. Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "Towards learning a universal non-semantic representation of speech," *arXiv preprint arXiv:2002.12764*, 2020.