A Sequential Audio Spectrogram Transformer for Real-Time Sound Event Detection

Takezo Ohta[1,2], Yoshiaki Bando[2], Keisuke Imoto[2,3], Masaki Onishi[2]
[1] Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
[2] National Institute of Advanced Industrial Science and Technology, Japan
[3] Department of Information Systems Design, Doshisha University, Japan

***Abstract*—In this paper, we propose an audio spectrogram transformer (AST) for sequential inference and evaluate its real-time performance. ASTs are pre-trained in a self-supervised manner, such as masked autoencoding, and the pre-trained models are well-performing in sound event detection. However, the existing architectures are designed for offline inference, wherein the entire signal serves as the input, and are unsuitable for sequential inference as they require the input sequence to be split into short chunks. In this study, we design a sequential AST based on a memory token (MT-AST) and its training method and conduct comprehensive experiments regarding the chunk length configuration. Specifically, we extend the offline AST with special tokens that memorize past signal information so that the network avoids repetitive inference of the same signal. While our model has limited inference capability, we train it using knowledge distillation from BEATs, a large-scale pre-trained model. Compared to the offline architecture, our model achieved higher performance by pre-training with AudioSet and fine-tuning for the URBAN-SED and DESED datasets. In addition, we conducted experiments to investigate the input chunk length considering performance-latency trade-offs and revealed the optimal configurations. We revealed that our model requires at least one extra second of input to maintain the performance.**

***Index Terms*—Sound event detection, audio spectrogram transformer, sequential inference**

I. Introduction

Sound event detection (SED) involves predicting sound event labels and their temporal activation observed in an input mixture signal. SED forms the foundation for applications such as surveillance systems that detect anomalies [1] and autonomous robots that respond appropriately to inputs from the environment [2]. A typical approach is to train a neural network that predicts the posterior probabilities of sound events for each time frame in a supervised manner [3]–[5].

Most existing SED systems are designed to work in an offline manner and process the entire signal as input. The convolutional recurrent neural network (CRNN) [6], for example, has widely been trained for SED. It combines a CNN to extract local features and a bidirectional RNN to extract sequential features. Transformer-based architectures [4], [5] have significantly improved prediction performance by extracting global features based on self-attention mechanisms. The audio spectrogram transformer (AST) [7], characterized by its pure attention-based architecture, splits an input spectrogram into patches and is utilized for SED after large-scale pre-training [5]. BEAT [8], an extension of AST, is trained on AudioSet [9] in a self-supervised manner to predict discrete tokens. This model effectively improved detection performance in the DCASE2023 Challenge Task 4 [10], [11].

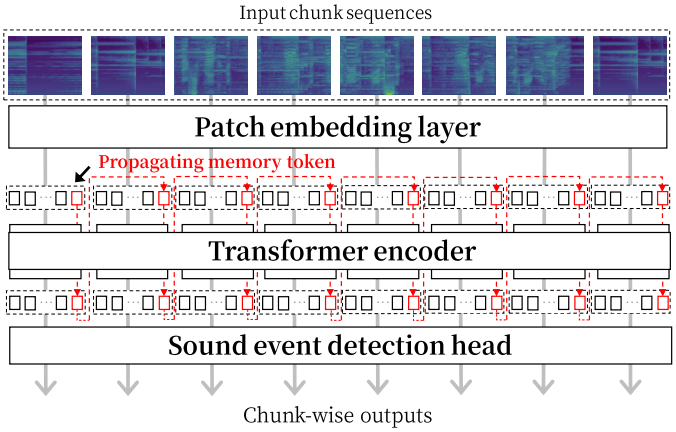


Fig. 1: Chunk wise inference based on memory token

Although offline models have been extensively investigated, real-time inference, considering latency, is essential for real-life applications. A naive approach is to split the input signal into short-time chunks for sequential inference. For example, the CNNs [12] can be used for sequential inference by sliding their receptive fields chunk-by-chunk. For discriminating long events and considering acoustic scenes, the AST is expected to effectively detect sound events in a sliding window manner with overlapping input chunk sequences. While the AST utilizes information from the longer input sequence, this leads to repeated inferences of the same chunk, which is computationally inefficient. Also, it is important to process look-ahead chunks with the current chunk for improving detection performance. The required length, which is critical for a tradeoff between performance and latency, remains ambiguous.

In this study, we propose a sequential AST utilizing memory tokens [13] (MT-AST) for chunk-wise inference (Fig. 1) and its training method based on knowledge distillation. Our MT-AST propagates memory tokens across chunks to utilize past inference information without repeated inference. This study is the first to extend the offline AST for sequential inference by introducing memory tokens that accumulate information

This study was partly supported by the New Energy and Industrial Technology Development Organization.

on past inferences into SED. In addition, we pre-train the MT-AST based on knowledge distillation from a BEAT model instead of directly training the model in a supervised manner. Thus, despite its limited inference capability, the MT-AST can acquire knowledge close to that of large-scale models.

The main contributions of this study are to develop a sequential AST for SED using large-scale datasets and to experimentally evaluate its performance and computational complexity. We pre-train the MT-AST on AudioSet [9] to achieve performance comparable to that of the teacher models. Moreover, we evaluated the trained models in various settings regarding the chunk length to clarify optimal configurations for performance-latency trade-offs. While the large-scale pre-trained ASTs have achieved high performance for offline inference, they have not yet been attempted for sequential inference of SED. This study challenges applying transformer-based pre-trained models, such as BEATs, to real-life applications. In the experimental evaluation, the models were transferred to SED on URBAN-SED [14] and DESED [6]. The performance of the proposed model was confirmed to be comparable to that of offline and supervised pre-trained models.

II. Related Work

This section overviews the existing offline architectures for SED and sequential processing tasks to position our method.

A. Offline architectures for SED

Offline architectures for SED are known to perform excellently owing to their context-aware global feature extraction [3]–[5]. The CRNN combines the CNN's local feature extraction with the RNN's temporal context modeling, achieving higher performance than when using each independently [3]. Extensions to the CRNN architecture, such as frequency dynamic convolution [15] and selective kernel attention [16], enable more flexible local feature extraction. These models are trained for higher performance in frameworks such as mean teacher [17] and pseudo-labeling [18], using a large-scale dataset [10]. Large-scale pre-trained embeddings of the transformer have also been reported to effectively enhance the performance of the existing CRNNs [8].

Various architectures based on transformers have been proposed for SED. Conformer [4], for example, combined a convolution layer with a self-attention mechanism, showing that attention-based global feature extraction performs well. AST-SED [5] extends the AST to handle overlapping events with high time resolution using the attention layer in the frequency axis and a high-resolution decoder. For pre-training the AST, the masked autoencoding [19] achieves state-of-the-art performance by fine-tuning for the SED [20], [21]. While most existing studies aim to design offline architectures, our study aims to design a transformer-based architecture that is efficient for sequential inference.

B. Sequential inference for real-time inference

Although deep neural networks perform well, they are also quite computationally complex, so various extensions for sequential processing have been proposed. The RNN, a sequence modeling structure, is inherently suitable for sequential processing and is used when real-time inference is considered [22]. Despite not having a sequential design, many studies have attempted to utilize CNN to use information from past signals because of their performance in various tasks [23], [24]. As a framework for memorizing previous signals, incremental inference using a part of the input signal is proposed [25]. In [26], a framework that utilizes future information for a specific interval was proposed, and experiments were conducted to verify the interval required to improve its performance. A study proposed caching all hidden-layer features and reducing past recalculations [27].

Sequential inference based on a transformer has also been studied to extract global features from a long signal. Transformer-XL [28] was designed to process the input sequences split into segments. There are examples of applying conformers to real-time sequential chunk processing [29], and the AST, which splits spectrograms into patches, has been applied in audio tagging [30]. The keys and values of the self-attention calculation were reused to avoid redundant recalculations of past signals. Unlike these approaches, we extend the AST to a recursive architecture with memory tokens for sequential inference.

III. MT-AST for Sound Event Detection

The proposed MT-AST employs memory tokens to leverage information from previous inferences and efficiently performs chunk-wise inferences. The MT-AST is pre-trained by knowledge distillation from the BEAT model, one of the most powerful pre-trained models. The model is then fine-tuned for the downstream task, SED, with a few audio recordings with strong labels. Pre-training and transfer learning were performed using a constant-length signal as input.

A. Sequential AST based on memory token

Our MT-AST processes an input spectrogram by splitting it into $P \times P$ patches, each of which is then flattened to a vector $\tilde{\mathbf{x}}_{tf} \in \mathbb{R}^{P^2}$, where t and f are the time and frequency indices of the patches, respectively. The input vectors $\tilde{\mathbf{x}}_{tf}$ are converted into input embeddings $\mathbf{x}_{tf} \in \mathbb{R}^D$ using a linear projection $\mathbf{A} \in \mathbb{R}^{D \times P^2}$ and a bias $\mathbf{b} \in \mathbb{R}^D$:

$$\mathbf{x}_{tf} = \mathbf{A}\tilde{\mathbf{x}}_{tf} + \mathbf{b}. \tag{1}$$

These embeddings are split into T patches in the time axis to form a chunk $\mathbf{X}^{(i)} \triangleq \{\mathbf{x}_{(iT)1}, \ldots, \mathbf{x}_{(iT+T)F}\} = \{\mathbf{x}_n^{(i)}\}_{n=1}^N \in \mathbb{R}^{N \times D}$, where i denotes the chunk index and $n = 1, \cdots, N$ is the index of the serialized patches in the chunk. Then, the AST inferences chunk sequence $\mathbf{X}^{(i:i+K)} \triangleq \{\mathbf{X}^{(i)}, \cdots, \mathbf{X}^{(i+K)}\}$ composed of the current chunk and K future chunks. To use information from past signals, L memory tokens $\mathbf{M}^{(i-1)} \triangleq \{\mathbf{m}_l^{(i-1)}\}_{l=1}^L \in \mathbb{R}^{L \times D}$ from the previous chunk's output are used with the input sequences to infer the current chunk. The encoder $\mathcal{F}$ takes these tokens as input and outputs the

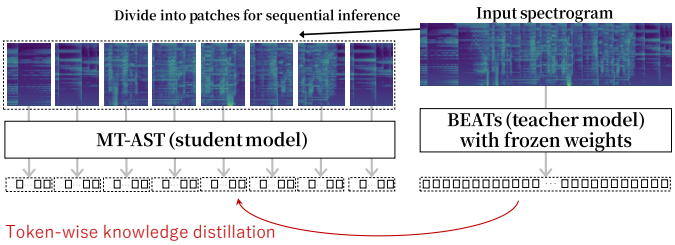


Fig. 2: Token-wise knowledge distillation from BEATs

embeddings $\mathbf{Z}^{(i:i+K)} \triangleq \{\mathbf{Z}^{(i)}, \cdots, \mathbf{Z}^{(i+K)}\} \in \mathbb{R}^{KN \times D}$ and the memory tokens $\mathbf{M}^{(i)}$ of the next chunk input:

$$\{\mathbf{Z}^{(i:i+K)}, \mathbf{M}^{(i)}\} = \mathcal{F}(\mathcal{P}(\{\mathbf{X}^{(i:i+K)}, \mathbf{M}^{(i-1)}\})), \quad (2)$$

where $\mathcal{P}$ is a layer adding the sinusoidal positional embedding.

B. Knowledge distillation from the BEATs model

For the pre-trained MT-AST, the embeddings of the large-scale pre-trained teacher model were distilled into the MT-AST using knowledge distillation. Specifically, the MT-AST is trained to approach its token-wise embeddings to those of BEATs. Let I be the total number of chunks for a training clip and $k = 1, \ldots, IN$ be the index of the output tokens corresponding to the k-th patch of the input spectrogram. The training objective $\mathrm{Loss}_{\mathrm{distil}}$ is defined as the average token-wise L2 loss between the output embeddings of the MT-AST $\{\mathbf{z}_i^{(s)}\}_{i=1}^{IN}$ and those of BEATs $\{\mathbf{z}_i^{(t)}\}_{i=1}^{IN} \in \mathbb{R}^D$:

$$\mathrm{Loss}_{\mathrm{distil}} = -\frac{1}{2}\sum_{i=1}^{IN} ||\mathbf{z}_i^{(s)} - \mathbf{z}_i^{(t)}||_2^2. \quad (3)$$

Learning token-wise embeddings instead of clip-wise embeddings enables the event class and temporal activation of the event with a resolution equivalent to the patch width. Distilled knowledge from the large model allows training the MT-AST with a small dataset when transferring it to SED.

C. Sequential AST for SED

The output embeddings of the chunks are converted into frame-wise embeddings to predict event activation. Specifically, the embeddings of an i-th chunk $\mathbf{Z}^{(i)}$ are reconstructed to be in 2D form embeddings $\mathbf{Z}_r^{(i)} \in \mathbb{R}^{F \times T \times D}$ and input to the frequency-wise transformer encoder (FTE) [5] to obtain D'-dimensional frame-wise class token $\mathbf{H}^{(i)} \in \mathbb{R}^{T \times D}$:

$$\mathbf{H}^{(i)} = \mathrm{FTE}(\mathbf{Z}_r^{(i)}). \quad (4)$$

The output layer $\mathcal{H}$ then converts $\mathbf{H}^{(i)}$ into frame- and class-wise predictions $\hat{\mathbf{Y}}^{(i)} \in \mathbb{R}^{T \times C}$ of acoustic events occurring in the i-th chunk:

$$\hat{\mathbf{Y}}^{(i)} = \mathcal{H}(\mathbf{H}^{(i)}), \quad (5)$$

where C denotes the number of sound event classes and the output layer $\mathcal{H}$ consists of two linear layers.

D. Transfer learning for SED

In training SED models, acoustic signal units of a certain length were used as the input. The model obtains the predictions for a whole clip $\hat{\mathbf{Y}} \in \mathbb{R}^{TI \times C}$ by arranging the chunk-wise predictions $\hat{\mathbf{Y}}^{(i)}$ in temporal order:

$$\hat{\mathbf{Y}} \triangleq \{\hat{\mathbf{Y}}^{(0)}, \ldots, \hat{\mathbf{Y}}^{(I)}\}. \quad (6)$$

The cross-entropy loss $\mathrm{Loss}_{\mathrm{sed}}$ is calculated from the obtained output and ground truth using the strong label $\mathbf{Y} \in \mathbb{R}^{TI \times C}$:

$$\mathrm{Loss}_{\mathrm{sed}} = -\sum_{t=1}^{TI}\sum_{c=1}^{C} y_{tc}\log(\hat{y}_{tc}) + (1 - y_{tc})\log(1 - \hat{y}_{tc}), \quad (7)$$

where $\hat{y}_{tc}$ and y_{tc} are the elements of $\hat{\mathbf{Y}}$ and $\mathbf{Y}$, respectively.

IV. Experimental Evaluation

We pre-trained the MT-AST on AudioSet [9] and transferred it to SED using two datasets, URBAN-SED [14] and DESED [6]. We evaluated the detection performance and the computation times for inference on an edge device.

A. Datasets

Knowledge distillation was conducted using the audio clips listed in AudioSet. We collected 1.77M clips from YouTube, which are approximately 85% of all clips listed in AudioSet. The dataset comprised various clips, including speech, music, and environmental sounds. Most clips were 10 seconds long, while some were less than 10 seconds. We used only clips that were 10 seconds long for training and validation. We used 1.6M videos (4,500 hours) for training and 1.6K videos (430 hours) for validation.

SED performance was evaluated using two public datasets: URBAN-SED [14] and DESED [6]. URBAN-SED provides synthetic mixture signals of urban soundscapes generated using a soundscape simulator called Scaper [14]. The dataset contains 16.7, 5.6, and 5.6 hours of 10-second signals for the training, validation, and test sets, respectively. They have annotations of ten event classes, including air conditioners, car horns, and sirens. DESED consists of synthetic mixture signals and real recordings in a domestic environment. The dataset contains unlabeled, weakly labeled, and strongly labeled clips. For simplicity, we used only the strongly labeled subset comprising 36, 3, and 2 hours of 10-second signals for the training, validation, and test sets, respectively. They have annotations of ten event classes, including dogs, dishes, and running water. All audio signals were resampled to 16 kHz, and the time resolution of the labels was set to 160 ms.

B. Experimental conditions

The MT-AST consisted of 12 Transformer encoder layers and 12 attention heads. The dimension of the hidden embeddings was 768. The number of memory tokens L was set to 20. As the input signal, a 128-bin ($F = 128$) log-Mel spectrogram was obtained using a short-time Fourier transform with a frame size and hop length of 400 and 160 samples, respectively. In addition, we examined the CRNN (DCASE CRNN) used as the baseline for DCASE2023 as our baseline.

The models were pre-trained using an AdamW optimizer [31] with a learning rate of 2.0×10^{-4} and weight decay of 1.0×10^{-5}. The training was performed for 4×10^5

TABLE I: Detection performance for URBAN-SED dataset

Method	Pre-training method	PSDS1	PSDS2	F1 [%]
DCASE CRNN		0.333	0.462	42.8
AST	strong supervision	0.186	0.313	28.7
AST	weak supervision	0.239	0.355	32.9
AST	BEATs distillation	0.354	0.486	48.3
MT-AST	strong supervision	0.242	0.431	33.0
MT-AST	weak supervision	0.276	0.477	36.0
MT-AST	BEATs distillation	0.377	0.518	48.3

TABLE II: Detection performance for DESED dataset

Method	Pre-training method	PSDS1	PSDS2	F1 [%]
DCASE CRNN		0.351	0.606	51.8
AST	strong supervision	0.215	0.498	35.5
AST	weak supervision	0.242	0.608	50.1
AST	BEATs distillation	0.313	0.632	57.7
MT-AST	strong supervision	0.196	0.514	32.1
MT-AST	weak supervision	0.241	0.520	39.8
MT-AST	BEATs distillation	0.310	0.696	53.3

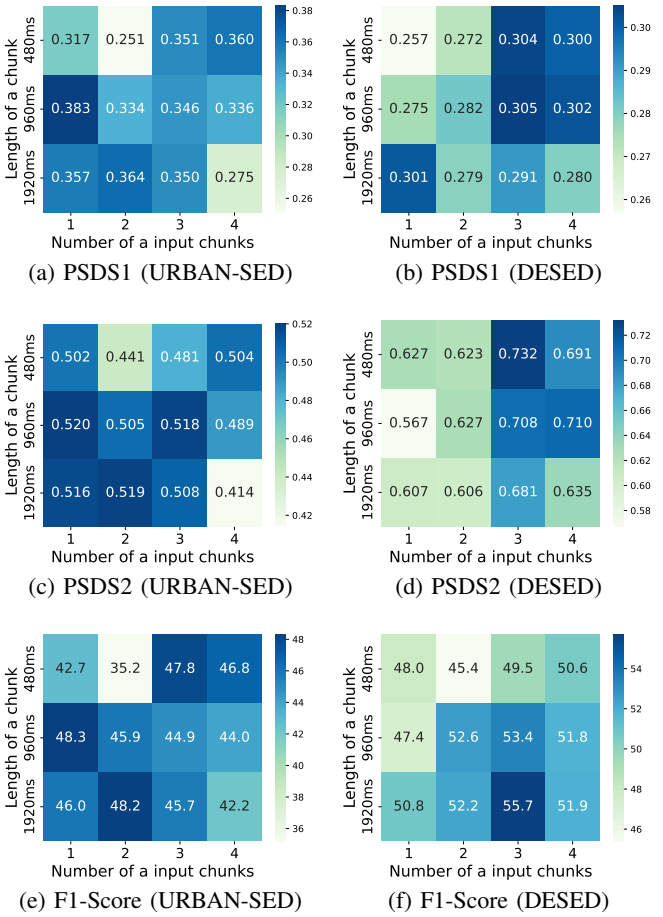
updates with a batch size of 256. The number of epochs for the pre-training was set at 200. We fine-tuned the pre-trained model using the Adam optimizer with a learning rate of 1e-4. The batch size was set to 64. We performed data augmentation techniques called time-frequency masking [32] and mixup [33]. The temporal resolution of the predicted labels was 160 ms, considering the stride of the patch embeddings. During DESED training, the frequency of real recordings appearing in a batch was increased 10 times. The hyperparameters were empirically determined.

We evaluated the detection performance using an event-based macro-averaging F1-score [34] and two polyphonic sound detection scores (the PSDS1 and the PSDS2) [35]. The PSDS1 evaluates detection scores focusing on the overlap of the model prediction and ground truth, whereas the PSDS2 focuses on the cross-triggers of sound events. We evaluated the model with the lowest validation loss and measured every 1000 update. The model prediction was smoothed using median filtering to improve event-based metrics. The ground truth temporal resolution was set to 160 ms to match the patch shift width. The time collar of the F1 score was set to 400 ms to consider the temporal resolution of the AST's output. We also investigated the differences due to chunk settings. For this experiment, we sampled 10% of the AudioSet for pre-training by random sampling because of computational resource limitations, and set the number of epochs to 100.

C. Evaluation of detection performance

The detection performance on the two datasets is listed in Tables I and II. In the URBAN-SED evaluation, our AST with memory tokens performed better than the offline model across all the metrics. In the DESED evaluation, the proposed model considerably surpasses the offline model for the PSDS2. The results show that the inference based on memory tokens utilizes past signal information better than the offline model. Furthermore, the results on both datasets show that knowledge distillation-based pre-training is stronger than supervised training based on annotated labels on all metrics. The MT-AST effectively learns BEATs embeddings through knowledge distillation, which is superior to supervised learning.

The heatmaps in Fig.3 show the detection performance at different settings of the chunk length and the number of input chunks (not including the past chunk). In URBAN-SED, the detection performance decreases when the input length is below 960 ms in the case where the chunk length is 480 ms, and the model could maintain the performance if the length



(a) PSDS1 (URBAN-SED) (b) PSDS1 (DESED)

(c) PSDS2 (URBAN-SED) (d) PSDS2 (DESED)

(e) F1-Score (URBAN-SED) (f) F1-Score (DESED)

Fig. 3: Effect of chunk length and number of input chunks on detection performance for URBAN-SED and DESED

of inputs is at least 960 ms. For DESED, the performance degrades considerably when the input length is below 960 ms. These results show that the input of approximately one second is necessary when prioritizing performance. In addition, no significant differences are observed in both datasets when the model input lengths are the same. This indicates that the memorization capability of the memory token does not correlate with the chunk length.

D. Evaluation of real-time inference performance

Table 3 shows the results of measuring the inference times on a device equipped with an Intel Core i7 CPU with 8 Core and 3.6 GHz baseline clock and 32 GB memory for

TABLE III: Elapsed times for inferencing a 10-second signal with different chunk lengths.

		lengths of a chunk [ms]		
		480	960	1920
Method	AST	6.69	3.72	2.11
	MT-AST (K=1)	0.87	0.74	0.59
	MT-AST (K=2)	1.10	1.01	0.89
	MT-AST (K=3)	1.48	1.17	0.91
	MT-AST (K=4)	1.68	1.48	1.17

the offline AST and the proposed MT-AST with different settings. Using a 10-second audio input, we averaged the time for the model to detect the signal over 500 inferences. Compared with the offline AST, the MT-AST considerably reduced the inference time, directly affecting latency. This result shows that using memory tokens effectively achieved low complexity inferences rather than inputting past signals. The memory tokens successfully expand the receptive field without increasing computational complexity.

V. Conclusion

We proposed a sequential AST for real-time SED and its training framework and evaluated the proposed method through experiments in various settings. We extended the AST to sequential inference using tokens that memorize past signal information and experimentally confirmed that it outperforms the offline method in terms of performance. We pre-trained the sequential AST using knowledge distillation based on token-wise embeddings of BEATs as a teacher. Experiments with various chunk settings provided insights into the relationship between performance and computational complexity. Future work includes implementing a real-time inference system on a resource-constrained edge device.

References

[1] C. Clavel *et al.*, "Events detection for an audio-based surveillance system," in *Proc. of IEEE International Conference on Multimedia and Expo*, 2005, pp. 1306–1309.

[2] X. Li *et al.*, "On-line sound event detection and recognition based on adaptive background model for robot audition," in *Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 1089–1094.

[3] E. Cakır *et al.*, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[4] K. Miyazaki *et al.*, "Conformer-based sound event detection with semi-supervised learning and data augmentation," *Proc. of workshop Detection and Classification Acoustic Scenes and Events (DCASE)*, vol. 1, pp. 100–104, 2020.

[5] K. Li *et al.*, "Ast-sed: An effective sound event detection method based on audio spectrogram transformer," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[6] N. Turpault *et al.*, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. of workshop Detection and Classification Acoustic Scenes and Events (DCASE)*, 2019, pp. 1–5.

[7] Y. G *et al.*, "AST: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[8] C. S *et al.*, "BEATs: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[9] G. JF *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[10] K. JW *et al.*, "Semi-supervsied learning-based sound event detection using freuqency dynamic convolution with large kernel attention for dcase challenge 2023 task 4," *arXiv preprint arXiv:2306.06461*, 2023.

[11] M. Chen *et al.*, "DCASE 2023 challenge task4 technical report," Detection and Classification Acoustic Scenes and Events (DCASE), Tech. Rep., 2023.

[12] E. Cakir *et al.*, "Filterbank learning for deep neural network based polyphonic sound event detection," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3399–3406.

[13] B. MS *et al.*, "Memory transformer," *arXiv preprint arXiv:2006.11527*, 2020.

[14] J. Salamon *et al.*, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. of Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[15] H. Nam *et al.*, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," *arXiv preprint arXiv:2203.15296*, 2022.

[16] K. JW *et al.*, "Semi-supervised learning-based sound event detection using frequency-channel-wise selective kernel for DCASE challenge 2022 task 4," *Proc. of workshop Detection and Classification Acoustic Scenes and Events (DCASE)*, 2022.

[17] X. Zheng *et al.*, "An effective perturbation based semi-supervised learning method for sound event detection." in *Proc. of INTERSPEECH*, 2020, pp. 841–845.

[18] S. Park *et al.*, "Self-training for sound event detection in audio mixtures," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 341–345.

[19] H. K *et al.*, "Masked autoencoders are scalable vision learners," in *Proc. of IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022, pp. 16 000–16 009.

[20] K. Li *et al.*, "Li USTC team' s submission for DCASE 2023 challenge task4a," Detection and Classification Acoustic Scenes and Events (DCASE), Tech. Rep., 2023.

[21] N. Shao *et al.*, "Fine-tune the pretrained ATST model for sound event detection," *arXiv preprint arXiv:2309.08153*, 2023.

[22] Y. Luo *et al.*, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[23] S. Bai *et al.*, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[24] A. Pandey *et al.*, "Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6875–6879.

[25] K. Wilson *et al.*, "Exploring tradeoffs in models for low-latency speech enhancement," in *Proc. of International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 366–370.

[26] M. Romaniuk *et al.*, "Efficient low-latency speech enhancement with mobile audio streaming networks," *arXiv preprint arXiv:2008.07244*, 2020.

[27] G. Stefański *et al.*, "Short-term memory convolutions," *arXiv preprint arXiv:2302.04331*, 2023.

[28] Z. Dai *et al.*, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[29] S. Chen *et al.*, "Continuous speech separation with conformer," in *Proc. IEEE ICASSP*. IEEE, 2021, pp. 5749–5753.

[30] H. Dinkel *et al.*, "Streaming audio transformers for online audio tagging," *arXiv preprint arXiv:2305.17834*, 2023.

[31] I. Loshchilov *et al.*, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, pp. 1–8, 2017.

[32] D. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, pp. 1–6, 2019.

[33] H. Zhang *et al.*, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, pp. 1–13, 2017.

[34] A. Mesaros *et al.*, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, pp. 1–17, 2016.

[35] Ç. Bilen *et al.*, "A framework for the robust evaluation of sound event detection," in *Proc. of IEEE ICASSP*, 2020, pp. 61–65.