Interspeech 2024
1-5 September 2024, Kos, Greece

Streaming Audio Transformers for Online Audio Tagging

Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, Yujun Wang, Bin Wang

AI Lab, Xiaomi Corporation, China

{dinkelheinrich,yanzhiyong, wangyongqing3, zhangjunbo1, wangyujun,wangbin11}@xiaomi.com

Abstract

Transformers have emerged as a prominent model framework for audio tagging (AT), boasting state-of-the-art (SOTA) performance on the widely-used Audioset dataset. However, their impressive performance often comes at the cost of high memory usage, slow inference speed, and considerable model delay, rendering them impractical for real-world AT applications. In this study, we introduce streaming audio transformers (SAT) that combine the vision transformer (ViT) architecture with Transformer-Xl-like chunk processing, enabling efficient processing of long-range audio signals. Our proposed SAT is benchmarked against other transformer-based SOTA methods, achieving significant improvements in terms of mean average precision (mAP) at a delay of 2s and 1s, while also exhibiting significantly lower memory usage and computational overhead.
Index Terms: Audio Tagging, Vision Transformer, Streaming inference, Online inference

1. Introduction

Audio tagging (AT) is a task that aims to label specific audio content into a fixed set of sound event classes, e.g., dog barking or people speaking. Applications of AT systems include aid for the hearing impaired, smart cities and homes [1] and general monitoring of sounds [2]. More recently AT systems have found applications on smartphones and smart speakers as a hearing aid for the needy. The transformer model, originally introduced in [3], which uses self-attention as its core building block, has become a popular method to achieve excellent performance for AT, however, the deployment of transformer architectures in real-world scenarios has been largely neglected. Previous works using Vision Transformer (ViT) based models such as in [4, 5, 6, 7, 8, 9, 10, 11] are optimized towards *offline* usage with a *global* context of 10s. Unfortunately, this approach results in a model response time (delay) of at least 10s. In our work, we define delay as the amount of data that a model needs to process before generating an output. Many transformer architectures in AT suffer from a high memory requirement due to their quadratic self-attention complexity, which depends on the amount of data processed at once (10s).

However, real-world applications are *online*, meaning that a model needs to return results as quickly as possible with a minimal delay while having access to a *limited* context (i.e., 1s). Although one may easily enable "online" inference by recomputing a 10s audio segment every e.g., 1s, this practice is inefficient, particularly when leveraging large transformer-based models [4]. To address this challenge, streaming inference algorithms aim to compute outputs efficiently without the need for recomputation by leveraging caching of previous results. This work solely focuses on optimising transformer-based models towards streaming inference, since traditionally used 2-dimensional convolutional neural networks (CNNs) are hard to make streamable [12].

We point out three essential prerequisites of AT models for real-world deployment, namely: (I) A minimal delay in terms of data necessary to output a label, typically on the order of 1-2 seconds. (II) A small memory footprint and low computational complexity. (III) Robust and reliable performance. While there exist many works in literature that tackle the problem of low delay [13], reducing memory footprint [7, 14] and improving performance [15, 4, 6], no comprehensive investigation has yet tackled all three issues. Thus, this work proposes **s**treamable **a**udio **t**ransformers (SAT), aimed at real-world usage of transformers for AT. Our motivation for this work is twofold. Firstly, it would improve compatibility between AT models and other audio subfields that are streaming, such as automatic speech recognition [16, 17], keyword spotting [18, 19] and source separation [20, 21]. Secondly, when deployed on stationary hardware like smart speakers, SAT models could act as an anomaly detector for long reoccurring sounds, differentiating between harmless and potentially harmful events, such as a single beep from a fire alarm versus continuous beeping. As we empirically demonstrate in this work, standard AT models struggle to continuously predict sound events (Section 5.4). Our contributions are: (I) We experiment with three standard-sized ViT models (Tiny, Small, Base), and optimize the training baseline for AT, aiming to reduce their memory consumption and decrease their floating-point operations per second (Flops). (II) Based on those three models, we introduce streamable (SAT) variants, denoted as SAT-T (Tiny), SAT-S (Small) and SAT-B (Base). We compare these models with other transformers in the literature and find significant performance improvements for short-delay inference.

2. Vision Transformers for Audio Tagging

Transformers were first proposed for machine translation in [3] and quickly became the state-of-the-art (SOTA) approach within the field of natural language processing (NLP) and later [22] the Vision Transformer (ViT) has been proposed as an adaption to the computer vision domain. Then, ViT-based transformers were used in AT, where images were replaced with two-dimensional spectrograms [4, 23]. The core idea of the ViT framework is the patchification operation, where an input spectrogram $\mathbf{S} \in \mathbb{R}^{F_{spec} \times T_{spec}}$ is first split into N non-overlapping patches of dimension d via a 2-dimensional convolution with a kernel-size P and stride P as:

$$\mathbf{X} = \text{Conv2D}(\mathbf{S}, P, P) = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}. \quad (1)$$

Then these resulting features (tokens) denoted as $\mathbf{X} \in$

10.21437/Interspeech.2024-242

$\mathbb{R}^{d\times N}$ are fed into a transformer model, consisting of a self-attention mechanism (Equation (2)) followed by a feed-forward neural network. Single-headed self-attention computes a similarity measure between linear transformations of the input $\mathbf{X}$ as:

$$\mathbf{Y} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \quad (2)$$

where $\mathbf{K} \in \mathbb{R}^{N\times d}, \mathbf{Q} \in \mathbb{R}^{N\times d}$ and $\mathbf{V} \in \mathbb{R}^{N\times d}$ are the key, query and value matrices obtained by a linear transformation $\mathbf{W}_j \in \mathbb{R}^{d\times d}, j \in \{K, Q, V\}$ of the same input $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\mathbf{Y} \in \mathbb{R}^{N\times d}$ is the output.

Self-attention scales quadratically for both memory and computation cost in regard to the input sequence length N. For ViT based approaches, the sequence length N consists of time and frequency tokens, thus our goal in this work is to reduce the frequency dimension and restrict the number of time frames accessible to the model.

3. Streaming Transformers

Transformer-XL-like architectures address the length limitation by performing a chunk-wise sequence processing. Specifically, the input $\mathbf{X} \in \mathbb{R}^{d\times N}$ of length N is first split into a series of S chunks, each of a fixed length T. The model applies a chunk transformer encoder within each segment, consisting of a self-attention mechanism (Equation (2)) followed by a feed-forward neural network. However, the self-attention mechanism in each chunk also attends to the hidden states from the previous chunk, using a recurrence mechanism, which allows the memory to scale linearly with respect to N. This recurrent mechanism helps the model to attend to past predictions and enlarge its context. The keys $\mathbf{K}_c \in \mathbb{R}^{d\times T}$ and values $\mathbf{V}_c \in \mathbb{R}^{d\times T}$ for chunk c, are concatenated with their previous hidden outputs as:

$$\tilde{\mathbf{K}}_c = [sg(\mathbf{K}_{c-1}) \parallel \mathbf{K}_c], \tilde{\mathbf{V}}_c = [sg(\mathbf{V}_{c-1}) \parallel \mathbf{V}_c], \quad (3)$$

where $sg(.)$ is a stop-gradient operation and $[\cdot \parallel \cdot]$ is the concatenation operation over the (time) dimension T. The stop-gradient operation is necessary to avoid vanishing/exploding gradients and this operation is done for each layer $l = 1, \ldots, L$. Note the length T_c of $\mathbf{K}_{c-1} \in \mathbb{R}^{d\times T_c}$ might differ from $\mathbf{K}_c$. Throughout our research, we conducted experiments with $T_c > T$; however, we observed no performance improvements. This lack of improvement can be attributed to the spontaneous characteristics of audio signals in Audioset. For instance, consecutive chunks may contain the same sound event, whereas chunks that are distant from each other may not share common sound events.

4. Experiments

4.1. Datasets

Our training and evaluation dataset is Audioset [24], which mainly contains 10-second long audio clips. We collected 1,904,746 training samples (5200 h) and 18,299 evaluation samples (50 h) sampled at 16 kHz. While our baseline models are trained and evaluated on the full context of an audio clip (10s), the proposed streaming transformers are mismatched between training (1s and 2s) and evaluation (10s). Thus, during evaluation, for clips longer than the chunk length, we split the input into equally sized chunks, then feed these chunks into the model and average all output scores.

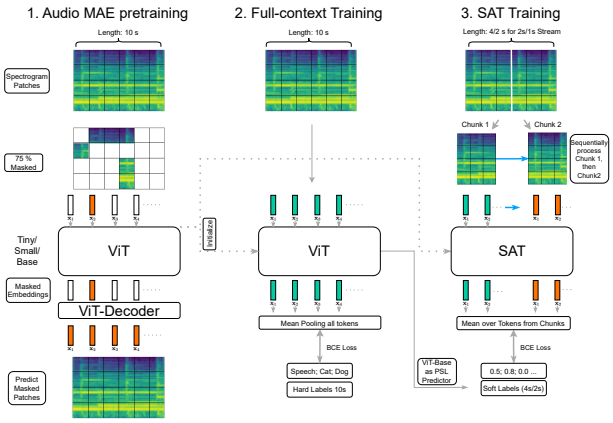

Figure 1: *The proposed training pipeline consists of three stages. First, pretraining using masked auto-encoders (MAE), second we use standard full-context training (10s clips) and third, our best model (ViT-B) is used to predict labels on a fine scale for SAT training.*

4.2. Training Pipeline

Our training pipeline can be summarized into three steps (see Figure 1). To achieve competitive performance, we first pretrain all our transformer models via the masked autoencoder (MAE) paradigm [25]. Second, we finetune the pretrained transformer model on Audioset using full-context training (10s). Third, pseudo strong labels (PSL) [13] are used to predict soft labels on a target length depending on the required delay and train each SAT model with these (soft) labels.

4.3. Setup

The present study employs three widely recognized ViT-based architectures, namely ViT-Tiny (ViT-T), ViT-Small (ViT-S), and ViT-Base (ViT-B) [22]. Each of these models comprises a $L = 12$ layer transformer with an embedding dimension d and a number of heads N_h, (d, N_h) set to $(192, 3)$, $(384, 6)$ and $(768, 12)$ for ViT-T/S/B, respectively. This study aims to explore the practical application of attention-based models, prioritizing factors such as inference speed and memory usage over marginal gains in performance and thus we optimize the training pipeline to use as few resources as possible. In contrast to previous works [4, 5, 6, 10, 11], we use an initial batch-norm to normalize each Mel-filterbank independently. Furthermore, we employ 64-dimensional banks at a 16 kHz sampling rate, extracted within a 32 ms window and a 10 ms hop instead of the conventional 128-dimensional banks at 32 kHz. This significantly improves inference speed and lowers memory usage. Also, we train with an 8-bit Adam optimizer [26] to further conserve memory. As a result, the ViT-B model only requires 6 GB of memory during training with a standard batch size of 32. Therefore, it is feasible to train our models on a single graphics processing unit (GPU) without the need for large computation clusters. The neural network back-end is implemented in Pytorch [27] and the source code with pretrained checkpoints is publicly available[1]. We further provide details regarding each stage of the training process.

[1] https://github.com/RicherMans/SAT

Self-supervised pretraining We pretrain three transformers, one for each size using a vanilla MAE [25, 23] approach. In the MAE framework, 75% of the patches (see Equation (1)) from a spectrogram $\mathbf{S}$ are removed and forwarded through a ViT encoder. Then a decoder is tasked to reconstruct the masked patches. We append a ViT decoder with $d = 512, N_h = 8$, where $L = 8$ for ViT-S/B and $L = 4$ for ViT-T. After training, the decoder is removed and only the encoder is kept, which serves as the pretrained checkpoint for full-context and SAT training. We train on a single GPU with a batch size of 64, a starting learning rate of 2e-4, a linear warmup of 3 epochs and a final learning rate of 2e-5 using a half-cycle cosine decay [28]. The ViT-B model is pretrained for 40 epochs, while the ViT-T/S models run for 50 epochs.

Training of baseline Audioset models For all models, we use time- and frequency-independent absolute learnable position embeddings [11]. Further, we employ random shifting, volume gain, and polarity inversion as augmentation methods in the waveform domain and Specaugment [29], which masks 192 consecutive time-frames and 24 frequency bins, in the spectrogram domain. Mixup [30] in the waveform domain is also employed with $\alpha = 0.3$. Importance sampling based on the label frequency is applied to counter the long tail of infrequent classes, where one epoch in our training scheme consists of sampling 200 k audio clips without replacement. Further, we use patch-out [11] where we randomly drop 25% of the time- and 25% of the frequency patches and use stochastic depth regularization [31] with a probability of 0.1. The training objective is binary cross entropy (BCE) and runs for at most 180 epochs with learning rates of 5e-4/4e-4/1e-5 and a warmup of 60/55/10 epochs for ViT-T/S/B, respectively. For all experiments, we average the top-4 models with the highest mAP, the main evaluation metric.

4.4. Training of streaming transformers

SAT models' recurrent dependency on previous chunks during training requires looping over sequential segments, decreasing efficiency. We use PSL [13], where our best model (ViT-B) is tasked to predict fine labels on a 4/2s scale for the 2/1s delay streaming models respectively, such that SAT has access to a single past chunk. The SAT models are trained without augmentation and mixup, by randomly sampling 200 k crops from the dataset each epoch. Training runs for at most 250 epochs, with identical learning rate schedules as their full-context ViT counterparts. Evaluation on Audioset is done by feeding each model chunks of its respective delay and averaging these scores across each 10s clip.

5. Results

5.1. Baseline results for full-context audio tagging

Our primary objective is to validate our proposed full-context models by comparing them with other transformer-based works from the literature. We also provide flops, and peak memory requirements (without considering the model parameters) for running inference on a single sample. The results obtained from our training pipeline, as shown in Table 1, indicate that our models, ViT-T (44.2) and ViT-S (45.7), achieve SOTA results using significantly fewer tokens as compared to previous works. Even though our proposed ViT-B (47.4) underperforms against BEATs [6] (47.9), it also requires three times less memory and approximately half of the computational resources. Additionally, it is worth mentioning that while HTS-AT [7] consumes less memory and requires fewer Gflops, the model is strictly optimized for offline usage, which we will further elaborate on in Section 5.2.

Model	Size	#Token	$\hat{M}$	Gflops	mAP
PaSST-Tiny [8]	5.6	496	120 M	5.4	39.7
MS-Tiny [8]		496	120 M	5.4	40.3
Ours (ViT-T)		**256**	**52 M**	**2.7**	**44.2**
PaSST-S [8]	22	496	279 M	21	43.1
MS-Small [8]		496	279 M	21	44.2
AMAE-S† [23]		512	291 M	22	45.0
Ours (ViT-S)		**256**	**105 M**	**10**	**45.7**
AST† [4]	86	1212	2.2 G	202	45.9
KD-AST† [5]	86	1212	2.2 G	202	47.1
MS-Base [8]	86	496	575 M	83	47.1
PaSST [11]	86	512	562 M	85	47.1
AMAE-B† [23]	86	512	562 M	85	47.3
AMAE-L† [23]	304	512	1.5 G	300	47.4
HTS-AT [7]	31	1024	**171 M**	**14**	47.1
BEATs† [6]	90	496	620 M	100	**47.9**[2]
Ours (ViT-B)	86	**256**	210 M	42	47.4

Table 1: *Baseline results of ViT-based AT models with pretraining compared to other works. Input length in number of tokens is denoted by #token, and model size is in millions. $\hat{M}$ represents peak memory in bytes during a forward pass (excluding model parameters). Models with † require multiple GPUs. The best result for each category is in bold.*

5.2. Streaming Audio Transformers

The results are displayed in Table 2, where we compare them against our own baseline models as well as SOTA approaches such as AST [4], BEATs [6] and HTS-AT [7]. For the evaluation of these SOTA approaches, we utilize their respective publicly available checkpoints with a delay of 2s. This means that we provide each model with a chunk of length 2s and average the scores, which aligns with how we evaluate our streaming models. Results from the experiment show that all models perform worse when evaluated with a 2s delay compared to full-context evaluation (see Table 1). This is primarily due to the limitation of accessing only a single chunk at a time, which restricts the models' ability to utilize future context, which is necessary for well-performing *offline* AT models. However, while AST (46.9 $\rightarrow$ 39.7), BEATs (47.9 $\rightarrow$ 38.7), and HTS-AT (47.1 $\rightarrow$ 5.2) all showed a significant drop in performance when evaluated with a delay of 2s compared to their respective full-context counterpart, our most efficient SAT-T model outperformed all baselines with significantly lower memory and flops requirements. This suggests that the other approaches may be tailored towards their target duration of 10s and struggle to be effective for real-world online evaluation. Our best SAT-B model with a 2s delay achieves an mAP of 45.1, which is 0.8 mAP points lower than full-context AST but uses 1.6% of AST's memory and 3.7% Gflops, while also having a significantly shorter delay of 2s. Further decreasing the delay to 1s decreases performance across all evaluated models, even though our proposed

[2]Our own evaluation of the public checkpoint reproduced 45.5. AST and HTS-AT are consistent with their reported results.

	Model	Strm?	#Token	$\hat{M}$	GFlops	mAP
2s delay	ViT-T	✗	48	7.6 M	0.5	39.1
	ViT-S		48	15 M	2.1	40.9
	ViT-B		48	30 M	8.2	41.5
	AST*		1212	2.2 G	202	39.7
	BEATs*		96	83 M	17.8	38.7
	HTS-AT*		1024	171 M	42	5.2
	SAT-T	✓	48/48	9 M	0.5	43.3
	SAT-S			18 M	2.1	43.4
	SAT-B			36 M	8.2	45.1
1s delay	ViT-T	✗	24	3.8 M	0.3	33.0
	ViT-S		24	7.5 M	1.1	34.9
	ViT-B		24	14 M	4.1	34.2
	AST*		1212	2.2 G	202	36.6
	BEATs*		48	83 M	17.8	35.2
	HTS-AT*		1024	171 M	42	2.4
	SAT-T	✓	24/24	4.3 M	0.3	40.1
	SAT-S			9 M	1.1	40.2
	SAT-B			16 M	4.1	41.4

Table 2: *Streaming Transformer results on Audioset using evaluation with a 2s and 1s delay. When streaming is used, #Token refers to T/T_c. "Strm?" indicates whether the model is streaming. $\hat{M}$ refers to the peak memory consumed (in bytes) during a forward pass of a single sample (with cache). * denotes evaluation from a public checkpoint. HTS-AT and AST pad their input to 10s, thus Gflops and $\hat{M}$ are unaffected.*

SAT-T/S/B models still significantly outperform the baselines in this condition.

5.3. Segment-level evaluation

To further validate the capability of SAT in accurately predicting chunk-level tags, we conducted evaluations using the strongly labeled Audioset evaluation set [32], which contains 381 matching labels with the weakly labeled Audioset used in this study. However, it is important to note that the labeling scheme differs between the two sets. For instance, the labeling scheme in the strongly labeled Audioset assigns labels such as "Speech" to either "Male" or "Female," resulting in significant performance discrepancies. We present the Segment-F1 (Seg-F1) and event onset-F1 scores [33], obtained by thresholding each model's output with a value of 0.5 at each respective chunk length. As the results in Table 3 indicate, our proposed SAT approach consistently outperforms other powerful transformer models in terms of segment-level performance on the strongly labeled Audioset.

5.4. Quantitative Analysis

A particularly useful application for streaming transformers is the (continuous) detection of long-duration sound events. However, to the best of our knowledge, there exists no public dataset that contains long audio samples that can simulate real-world audio streams. Thus, we collected two 10-minute water sound samples from YouTube[3] and evaluated different AT models as depicted in Figure 2. Two sets of experiments were conducted, one at a full-context (10s) scale, matching the delay of standard AT approaches, and one with limited-context (2s) inputs matching the resolution of our proposed SAT. Note that these two samples are monotonous and repetitive; in other words, they should be easily detected by a potent AT model with a consistently high probability over a prolonged amount of time. We compared our smallest model, SAT-T, with BEATs, HTS-AT, and AST, and found that SAT-T consistently outperformed the competitors in predicting the presence of water with high confidence and lower delay for all experimented settings. Further, due to its strict optimization to a 10s delay, HTS-AT is incapable to predict any presence of water, which is in line with the observations in Table 2.

[3]Accessible via `www.youtube.com/watch?v=VIDEO_ID`

Model	Strm?	2s Seg-F1	2s Onset-F1	1s Seg-F1	1s Onset-F1
ViT-T	✗	18.2	6.2	11.1	6.2
ViT-S		19.0	6.4	11.7	6.4
ViT-B		18.8	6.2	10.2	6.2
AST		13.7	4.7	10.5	4.7
BEATs		16.1	5.7	11.0	5.7
SAT-T	✓	28.9	8.6	24.2	7.3
SAT-S		30.6	9.0	25.3	7.1
SAT-B		30.1	8.4	22.3	8.3

Table 3: *Results for the segment and onset-based F1-score evaluation on the strongly labelled Audioset evaluation set using a threshold of 0.5 for 2s and 1s input segments. Models have been trained on the weakly labeled Audioset.*

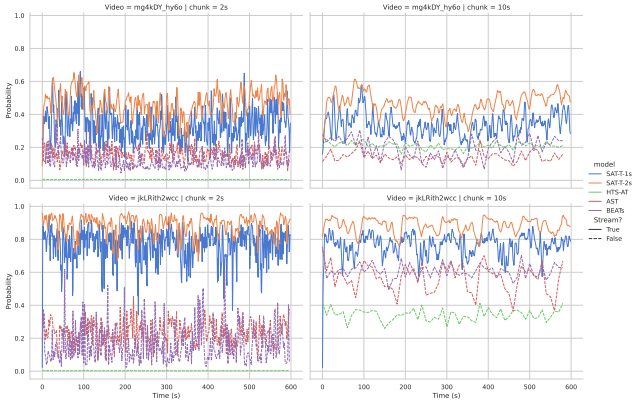

Figure 2: *Comparison of output probability scores between the baselines against the proposed SAT-T for a 10-minute long sound of water event. Samples (Top: `mg4kDY_hy6o`, Bottom: `jkLRith2wcc`) were taken from Youtube and evaluated using 2s (left) and 10s (right) chunks. Best viewed in color.*

6. Conclusion

This paper presents streaming transformers for AT, which have lower memory usage, faster response time, and can capture long-range sound events. The paper proposes three ViT-based models (SAT-T/S/B) with two streaming configurations for short-delay AT. The best model, SAT-B, achieves an mAP of 45.1 with a 2s delay, using 8.2 Gflops and 36 MB of memory during inference. The paper also demonstrates through a simulation that the proposed streaming transformers outperform other state-of-the-art methods in terms of long-term consistency.

7. References

[1] J. P. Bello, C. Mydlarz, and J. Salamon, *Sound Analysis in Smart Cities*. Cham: Springer International Publishing, 2018, pp. 373–397.

[2] W. Xiong, X. Xu, L. Chen, and J. Yang, "Sound-based construction activity monitoring with deep learning," *Buildings*, vol. 12, p. 1947, 11 2022.

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

[4] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. Interspeech 2021*, 2021, pp. 571–575.

[5] Y. Gong, S. Khurana, A. Rouditchenko, and J. Glass, "Cmkd: Cnn/transformer-based cross-model knowledge distillation for audio classification," *arXiv preprint arXiv:2203.06760*, 2022.

[6] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

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

[8] D. Chong, H. Wang, P. Zhou, and Q. Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

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

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

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

[12] G. Stefański, K. Arendt, P. Daniluk, B. Jasik, and A. Szumaczuk, "Short-term memory convolutions," *arXiv preprint arXiv:2302.04331*, 2023.

[13] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, and Y. Wang, "Pseudo strong labels for large scale weakly supervised audio tagging," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 336–340.

[14] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," *arXiv preprint arXiv:2211.04772*, 2022.

[15] 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.

[16] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6074–6078.

[17] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," *arXiv preprint arXiv:1909.05330*, 2019.

[18] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming Keyword Spotting on Mobile Devices," in *Proc. Interspeech 2020*, 2020, pp. 2277–2281.

[19] Y. Wang, H. Lv, D. Povey, L. Xie, and S. Khudanpur, "Wake word detection with streaming transformers," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5864–5868.

[20] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika, and A. Gruenstein, "VoiceFilter-Lite: Streaming Targeted Voice Separation for On-Device Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2677–2681.

[21] R. Rikhye, Q. Wang, Q. Liang, Y. He, and I. McGraw, "Closing the gap between single-user and multi-user voicefilter-lite," *arXiv preprint arXiv:2202.12169*, 2022.

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

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

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

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

[26] T. Dettmers, M. Lewis, S. Shleifer, and L. Zettlemoyer, "8-bit optimizers via block-wise quantization," *9th International Conference on Learning Representations, ICLR*, 2022.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library." Curran Associates, Inc., 2019, vol. 32, pp. 8026–8037.

[28] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[30] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[31] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 646–661.

[32] S. Hershey, D. P. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 366–370.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, p. 162, 2016.