COMPARISON OF LOW COMPLEXITY SELF-ATTENTION MECHANISMS FOR ACOUSTIC EVENT DETECTION

Tatsuya Komatsu and Robin Scheibler
LINE Corporation, Tokyo, Japan
E-mail: komatsu.tatsuya@linecorp.com

***Abstract*—We investigate and compare several low-complexity self-attention mechanisms applied to the problem of acoustic event detection. Self-attention has proved to be all that is needed to make leaps in several domains, but at a computational and memory cost quadratic in the length of the input sequence. This problem has been recently addressed by several works that reduced the complexity: linear attention, top-k attention, clustered attention, attention-free Transformer, and FNet. We replace the conventional self-attention block of an acoustic event detection model by these low complexity ones and evaluate the performance on Task 4 of the DCASE Challenge 2021. We find that at the cost of marginal performance drop the inference time was significantly sped up for sequences $30\,\mathrm{s}$ and longer. We conclude that for all practical purposes, one of these low-complexity attention mechanism should be used instead of the conventional one.**

I. Introduction

Understanding the various sounds around us is being actively researched for various applications such as robot audition, hearing assistance, and street surveillance [1], [2], [3], [4]. Acoustic event detection, which is the recognition and detection of environmental sounds, is a key technology for such applications. Acoustic event detection is undergoing remarkable progress due to the development of deep learning techniques and the construction of huge datasets such as Audioset [5]. The annual DCASE event with specialized workshops and competitions has also helped its development.

Acoustic event detection is a supervised classification technique that uses detection models for predefined event classes to detect which event class occurred where in the input speech. Much research has been conducted on how to build detection models, such as mel-frequency cepstrum coefficients and Gaussian mixture models based on speech recognition techniques [6], and factor decomposition models such as non-negative matrix factorization focusing on the overlap of sounds [7], [8], [9]. With the development of deep learning, lots of models, such as convolutional neural networks (CNNs) [10], long short-term memory (LSTM) [11], [12] and convolutional recurrent neural networks (CRNNs) [13], [14], have shown high detection performance. In particular, CRNN-based models have been employed as a powerful baseline method in many studies.

More recently, methods based on Transformer [15] have shown significant improvements over the CRNN-based models [16], [17], [18]. In particular, in the recent DCASE Challenges, methods based on Conformer [19], [20], a variant of Transformer, have shown top performance. Most of the Transformer-based methods are based on replacing the RNN part of CRNN-based methods with Transformer encoders. Transformer-based methods take the spectrum of the input audio, transform it into a sequence of high-dimensional representations by a CNN, and extract frame-level features based on global relations in the sequence by the Transformer encoder. The key to capturing the global relations within a sequence is the self-attention mechanism of the Transformer encoder. The self-attention mechanism projects the input sequence into three types of series: query, key, and value. Based on the projected query and key, the relationship between each frame is represented as a weight matrix, and the values of each frame are transformed based on the weight matrix.

Self-attention has proved to be all that is needed to make leaps in several domains, but at a computational and memory cost quadratic in the length of the input sequence. This problem has been recently addressed by several works that reduce the complexity [21], [22]. The linear attention method replaces softmax by the linear product of feature maps computed from the query and key matrices. The top-k attention [23] sparsifies the attention matrix by zeroing all entries but the k largest of each row. Clustered attention [24] applies K-means clustering to the query matrix and then attention only for the cluster centers. Attention-free transformer [25] replaces the full attention by a row-wise operation that requires less memory. It can be further simplified to have linear computational complexity too. Finally, FNet [26] drastically simplifies attention by replacing altogether the operation by a 2D discrete Fourier transform (DFT), followed by a row-wise 2-layers feed-forward network.

In this paper, we investigate and compare those low-complexity self-attention mechanisms applied to the problem of acoustic event detection.

II. AED with Transformer

A. Transformer

Let us consider acoustic event detection (AED) with C event classes. AED is a multi-label classification problem that takes spectrogram of the input audio $\mathbf{V} \in \mathcal{R}^{T \times F}$ and estimates the event classes $\mathbf{Y} \in \mathcal{R}^{T \times C}$ occuring in each time frame T, where T, C and F are the number of time frames, classes and feature dimensions, respectively.

Transformer-based model consists of the CNN layers as feature extractor followed by N stacked Transformer encoder layers. The CNN layers extract high-dimensional representation $\mathbf{X}_0 \in \mathbb{R}^{T\times D}$ from input spectrogram $\mathbf{V}$. The Transformer encoder layers takes the CNN output as input and transform the input based on the self-attention mechanism. The Transformer encoder consists of N stacked encoder layers, and each encoder layer consists of two sub-layers, a multi-head attention $\mathrm{MHA}(\cdot)$ and a position-wise feed-forward network $\mathrm{FFN}(\cdot)$. Input of each encoder layer is the output of the previous layer, where input and output are sequences of the same dimensions. Let the input and output of the n-th encoder layer be denoted $\mathbf{X}_{n-1} \in \mathbb{R}^{T\times D}$ and $\mathbf{X}_n \in \mathbb{R}^{T\times D}$, respectively. The operation of the n-th encoder layer is described as follows:

$$\mathbf{H}_n = \mathrm{MHA}(\mathbf{X}_{n-1}) \tag{1}$$

$$\mathbf{X}_n = \mathrm{FFN}(\mathbf{H}_n) \tag{2}$$

where $\mathbf{H}$ is a latent representation extracted by the multi-head attention (MHA), and $\mathrm{FFN}(\cdot)$ applies a nonlinear transformation to each time frame of the sequence $\mathbf{H}$. For stable training, each sub-layer has layer normalization [27] and residual connections [28]. The final output $\mathbf{X}_N$ of the N encoder layers is fed to the subsequent decoder.

Finally, $\mathbf{X}_N$ is transformed by the classifier into $\hat{\mathbf{Y}} \in \mathcal{T}^C$, estimation of $\mathbf{Y}$.

$$\mathbf{Y} = \mathrm{Classifier}\left(\mathbf{X}_N\right) \tag{3}$$

There are several ways to design a classifier, most of which use a full coupling layer and a sigmoid function, which is also used in this paper.

The parameter training of the network is generally based on a cost function consisting of sigmoidal cross-entropy.

$$\mathcal{L} = \mathrm{SCE}\left(\hat{\mathbf{Y}}, \mathbf{Y}\right) \tag{4}$$

In recent years, there has been a focus on weakly labeled learning, where there is no timestamps for occurrence of events. The weak labels indicate the presence or absence of an event in each audio clip, but not at what time the sound has occurred, so that this cost function cannot be used in weak label training. Other methods such as semi-supervised training like mean-teacher and domain adaptation are employed for the weak label training.

B. Self-attention mechanism

The $\mathrm{MHA}(\cdot)$ performs scaled dot-attention multiple times in parallel (i.e., multi-head) and aggregates the individual attention results. The scaled dot-product attention is formulated as an operation on three matrices; query $\mathbf{Q} \in \mathbb{R}^{T\times d_q}$, key $\mathbf{K} \in \mathbb{R}^{T\times d_k}$ and value $\mathbf{V} \in \mathbb{R}^{T\times d_v}$. In this paper, dimensions of all matrices are set to be the same, $d_q = d_k = d_v = d$. The similarity between $\mathbf{Q}$ and $\mathbf{K}$ is first calculated by inner product and normalization with the softmax function. Then, $\mathbf{V}$ is summarized as a weighted sum based on the similarity:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \tag{5}$$

In multi-head self-attention, the matrices query, key, and value of each attention-head h are obtained by applying linear transformations to a single input $\mathbf{X}$;

$$\mathbf{Q} : \mathbf{X}_h^{(q)} = \mathbf{X}\mathbf{W}_h^{(q)} \tag{6}$$

$$\mathbf{K} : \mathbf{X}_h^{(k)} = \mathbf{X}\mathbf{W}_h^{(k)} \tag{7}$$

$$\mathbf{V} : \mathbf{X}_h^{(v)} = \mathbf{X}\mathbf{W}_h^{(v)} \tag{8}$$

Using these matrices, Eq. 5 can be rewritten as follows:

$$\mathrm{SelfAttention}(\mathbf{X}) = \mathrm{softmax}\left(\frac{\mathbf{X}^{(q)}\mathbf{X}^{(k)^\top}}{\sqrt{d}}\right)\mathbf{X}_h^{(v)}, \tag{9}$$

$\mathbf{X}^{(q)}\mathbf{X}^{(k)^\top}$ can be interpreted as the similarity between each time frame in the sequence, and this information is the key for the Transformer to encode the input.

The computational and memory cost for the calculation of attention scores are on the order of quadratic in the length of the input sequence, $\mathcal{O}(T^2 d)$.

III. Low-Complexity Self-Attention Mechanisms

Recently, a number of techniques have been proposed to reduce the quadratic computational and/or memory requirements. We give here a brief introduction to the techniques investigated in this paper.

A. Linear Transformer

Linear attention relies on approximating the softmax operation of the attention by a linear dot-product of feature maps [29]. Specifically, the t-th row of the linear attention is given by

$$\frac{\phi(\mathbf{q}_t)^\top \sum_{t'=1}^{T} \phi(\mathbf{k}_{t'})\mathbf{v}_{t'}^\top}{\phi(\mathbf{q}_t)^\top \sum_{t'=1}^{T} \phi(\mathbf{k}_{t'})} \tag{10}$$

where $\mathbf{q}_t$, $\mathbf{k}_t$, and $\mathbf{v}_t$ are rows of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively. The function $\phi : d \rightarrow d'$ computes the feature map for a single row. Thus, the complexity of linear attention is $\mathcal{O}(Td^2)$, which is indeed linear in the sequence length, and computationally advantageous when $d \ll T$.

B. Top-k attention

The Top-k attention mechanism forces attention onto the most relevant parts of the sequence only by selecting the k largest entries in each row of the attention matrix [23], and ignoring all others. Originally motivated by a lack of focus of the conventional attention, improved training and testing times where also reported. While full computation of the attention matrix is still required, zeroing out most of its entries significantly reduces the cost of the subsequent (sparse) matrix-matrix product to $\mathcal{O}(kTd)$.

C. Clustered attention

Clustered attention brings down the complexity of attention to linear by clustering the rows of the query matrix $\mathbf{Q}$ with the K-means algorithm improved by locality-sensitive hashing [24]. Then, each of the time position of the input sequence only attends to one of C cluster centers and complexity is thus reduced to $\mathcal{O}(CTd)$. The resulting matrix $\mathbf{V}$ is built by repeating for all cluster members the attention value of its center. The complexity of the clustering operation is $\mathcal{O}(TCL + CBL + TdB)$ where L is the number of iterations of K-means, and B is the number of bits for the hashing. We note the constant overhead introduced.

D. Attention free Transformer

The attention free transformer (AFT) network replaces attention by a mechanism that can be applied time position wise to the input sequence [25]. This removes the need to compute and store the $T \times T$ attention matrix. Specifically, the t-row of the output is computed as

$$\sigma(\mathbf{q}_t) \odot \sum_{t'=1}^{T} \frac{\exp(\mathbf{k}_{t'} + b_{t,t'})}{\sum_{t'}^{T} \exp(\mathbf{k}_{t'} + b_{t,t'})} \odot \mathbf{v}_{t'}, \tag{11}$$

for $t = 1, \dots, T$, where $\mathbf{q}_t$, $\mathbf{k}_t$, and $\mathbf{v}_t$ are the rows of the matrices represented by the upper-case corresponding letter. A number T^2 of biases $b_{t,t'}$ is introduced. Equation 11 is refered to as AFT-full. While it avoids computing and storing the attention matrix $\mathbf{Q}\mathbf{K}^\top$, it introduces T^2 biases and its asymptotic computational complexity is the same as that of regular attention. Fixing $b_{t,t'} = 0$ for all positions yields a reduced complexity method termed AFT-simple with linear time complexity. The reduction is due to the softmax on the columns of $\mathbf{K}$ being computed only once.

E. FNet

FNet gets rid of the attention mechanism altogether and enforces a global mixing of the time steps via the DFT [26]. The input $\mathbf{X}$ is mapped to a matrix of the same size by a 2D DFT. Then, each row of the transformed matrix is passed through a 2 layers feed forward network. These two operations are repeated for several layers. This approach is very efficient because the DFT layer does not require any memory storage and can use the efficient fast Fourier transform algorithm with complexity $\mathcal{O}(Td(\log T + \log d))$. Furthermore, the complexity of the feed forward network is linear in the length of the sequence T, and its storage requirements independent of it.

IV. Experiments

Experiments were conducted to investigate the effects of different self-attention mechanisms on AED performance. All experiments were conducted using the same transformer-based model. The evaluation task was set as DCASE2021 task4 and the performance of weakly supervised AEDs by each method was compared. All the transformer parts were implemented using the Fast Transformer toolkit [29], [24], except for FNet.

TABLE I
Experimental conditions

Transformer parameters	
attention dim	144
# heads	4
eunits(=d_{ff})	576
# Encoders	4
dropout	0.2
Training settings	
Batch size	32
Iteration steps	20,000
Optimizer	Adam
Learning rate schduling	noam [15]
Learning rate scale	0.02
Warm up steps	4,000

A. Datasets

The dataset of DCASE2021 task4 [30], [31] was used in the experiments. The dataset is composed of 10 sec audio clips recorded in domestic environments or synthesized using Scaper [32] to simulate a domestic environment. The event classes to be detected consist of 10 classes that represent a subset of Audioset [5]. In our experiments, all clips were downsampled to 16 kHz beforehand.

There are 3 different splits of the training dataset: Labeled training set, Unlabeled in domain training set and Synthetic set with strong annotations.

Weakly-labeled training set
 1578 clips (2244 class occurrences) for which weak annotations.

Unlabeled in domain training set
 14412 clips without any annotations. The clips are selected such that the distribution per class (based on Audioset annotations) is close to the distribution in the labeled set.

Synthetic strongly labeled set
 10000 clips generated with the Scaper soundscape synthesis and augmentation library. The clips are generated such that the distribution per event is close to that of the validation set.

The validation set contains 1168 clips and is annotated with time-stamped labels. The domain of the validation set is the same as that of the weakly-labeled training set.

B. Experimental conditions

The input spectral features were 64-dimensional log-mel spectrogram extracted with window and hop lengths of 1024 and 323 points, respectively. The length of input frames was fixed to 496 frames (10 seconds). Audio clips shorter than 10 s were zero-padded. Prior to input to the network, the features were normalized for each bin as follows:

$$\bar{\boldsymbol{V}} = (\boldsymbol{V} - \bar{\boldsymbol{\mu}})/\bar{\boldsymbol{\sigma}}, \tag{12}$$

where $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\sigma}}$ represent mean and variance calculated from the training dataset, respectively.

TABLE II
EXPERIMENTAL RESULTS

	F1-score		Complexity	Memory	RTF (×1000)			# of params
	Segment-based	Event-based			10 sec clip	30 sec. clip	1 min. clip	(Transformer)
Transformer (full attention)	0.711	0.500	T^2d	T^2d	8.454	7.922	8.833	1003104
Top-K ($k = 16$)	0.710	0.462	T^2d	kTd	9.387	10.163	11.851	1,003,104
Top-K ($k = 32$)	0.700	0.473	T^2d	kTd	9.775	10.314	12.423	1,003,104
Top-K ($k = 64$)	0.715	0.495	T^2d	kTd	9.995	10.642	12.769	1,003,104
Clustered ($C = 16$)	0.702	0.459	CTd	CTd	10.132	8.760	8.626	1,003,104
Clustered ($C = 32$)	0.715	0.468	CTd	CTd	10.294	8.771	8.432	1,003,104
Clustered ($C = 64$)	0.715	0.469	CTd	CTd	11.188	9.334	8.919	1,003,104
AFT-full	0.697	0.457	T^2d	Td	14.968	27.567	47.719	1,527,392
AFT-Simple	0.693	0.458	Td	Td	7.884	7.657	7.621	1,003,104
Linear Transformer	0.704	0.473	Td^2	Td	7.880	7.141	7.391	1,003,104
FNet	0.681	0.411	$Td \log Td$	Td	7.534	7.031	7.429	668,736

Table I shows the network configuration and training parameters. We used a network with 7 CNNs layers and 4 Transformer layers for baseline, and only changed the part calculating the self-attention to compare the performance. The feature representation layer structure was the same as the CNN part of [33]. Note that the length of feature representation layer outputs was reduced to 1/8 by max-pooling. In this study, we set the Transformer encoder parameters as follows: attention units were set to 144 ($d = 144$), the number of heads was 4 ($h = 4$), internal position-wise feed-forward units were set to 576, and the dropout ratio was set to 0.2. The batch size was set to 64.

In order to use the weakly labeled audio clips, we used a method based on mean-teacher [34], which is a semi-supervised training method. The mean-teacher method is a widely used technique, such as in the baseline method and many other methods in the DCASE Challenges.

The evaluation metrics were segment-based and event-based F-scores. The calculation of each value was performed using the SED-Eval toolkit [35]. Segment-based F-scores are calculated whether events are correctly predicted in each segment, where the segment length is 1 second. Event-based F-scores are calculated based on the onset/offset of the predicted result. In this work, we set allowable length to prediction error of 200 ms for onsets and 200 ms / 20 % of event length for offsets. Model parameters of all methods were trained on 20,000 iterations and evaluated on their performance on the DCASE2021 task4 validation set.

We also measured the real time factor (RTF) during inference. The RTF was measured as the time it took to infer a series of 10 seconds, 30 seconds, and 1 minute in length, respectively. Inference for each length sequence was performed 100 times, and the average value was used as the RTF. All training is performed on a single V100 GPU and RTF is measured by decoding with a batch size 1 on a single Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz using a single thread.

C. Experimental Results

Table II shows the experimental results. Both the top-k and clustered attention show comparable performance to full attention. These methods focus on the sparsity and redundancy of attention masks and are able to represent attention masks with low memory cost. However, the complexity for the top-k attention is still on the same order as the full attention, since the full attention score needs to be computed once to perform top-k search. The clustered attention also shows slow inference speed for the 10 sec. clips due to the overhead for the clustering. Since the overhead is constant, the RTF decreased as the series gets longer. It is expected to show faster inference than full attention for longer sequences with $T \gg C$.

The Linear Transformer enables 16.3% faster inference for 1 min. sequences while degrading segment-based F-score by 0.98% and event-based F-score by 5.4% compare to the full-attention. The calculation of attention score is linear to the sequence length, and the speed does not decrease even for inference on long sequences.

Although FNet was not as good as the other methods in terms of performance, the inference speed was fast as the linear Transformer which shows 15.8% faster than the full attention. The most notable feature of FNet is the small number of parameters. FNet removes the attention operation and enforces a global mixing of the time steps via the DFT. Therefore, it is constructed with less than 70% of the number of parameters of other methods.

V. CONCLUSION

In this paper, we investigate several low-complexity attention mechanism methods for transformer-based acoustic event detection and evaluate their performance. In the experiments, the characteristics and performance of each method are discussed, and it is found that linear Transformer in particular can speed up the inference speed by 16.3% with small performance degradation. We conclude that for all practical purposes, one of these low-complexity attention mechanism can be used instead of the conventional one.

REFERENCES

[1] Johannes A Stork, Luciano Spinello, Jens Silva, and Kai O Arras, “Audio-based human activity recognition using non-markovian ensemble voting,” in *Proc. IEEE RO-MAN*, 2012, pp. 509–514.

[2] Keisuke Imoto, Suehiro Shimauchi, Hisashi Uematsu, and Hitoshi Ohmuro, “User activity estimation method based on probabilistic generative model of acoustic event sequence with user activity and its subordinate categories.,” in *Proc. INTERSPEECH*, 2013, pp. 2609–2613.

[3] Regunathan Radhakrishnan, Ajay Divakaran, and A Smaragdis, “Audio analysis for surveillance applications,” in *Proc. WASPAA*, 2005, pp. 158–161.

[4] Chloé Clavel, Thibaut Ehrette, and Gaël Richard, “Events detection for an audio-based surveillance system,” in *Proc. ICME*, 2005, pp. 1306–1309.

[5] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. ICASSP*, New Orleans, LA, 2017.

[6] Lode Vuegen, BVD Broeck, Peter Karsmakers, Jort Florent Gemmeke, Bart Vanrumste, and HV Hamme, “An MFCC-GMM approach for event detection and classification,” in *Proc. WASPAA*, 2013, pp. 1–3.

[7] Onur Dikmen and Annamaria Mesaros, “Sound event detection using non-negative dictionaries learned from annotated overlapping events,” in *Proc. WASPAA*. IEEE, 2013, pp. 1–4.

[8] Tatsuya Komatsu, Yuzo Senda, and Reishi Kondo, “Acoustic event detection based on non-negative matrix factorization with mixtures of local dictionaries and activation aggregation,” in *Proc. ICASSP*. IEEE, 2016, pp. 2259–2263.

[9] Tatsuya Komatsu, Takahiro Toizumi, Reishi Kondo, and Yuzo Senda, “Acoustic event detection method using semi-supervised non-negative matrix factorization with a mixture of local dictionaries,” in *IEEE AASP Challenge: DCASE2016*, 2016, pp. 45–49.

[10] Arseniy Gorin, Nurtas Makhazhanov, and Nickolay Shmyrev, “DCASE 2016 sound event detection system based on convolutional neural network,” in *IEEE AASP Challenge: DCASE2016*, 2016.

[11] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *Proc. ICASSP*. IEEE, 2016, pp. 6440–6444.

[12] Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, Takaaki Hori, Jonathan Le Roux, and Kazuya Takeda, “Duration-controlled LSTM for polyphonic sound event detection,” *IEEE TASLP*, vol. 25, no. 11, pp. 2059–2070, 2017.

[13] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” *IEEE TASLP*, vol. 25, no. 6, pp. 1291–1303, 2017.

[14] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen, “Sound event detection using spatial features and convolutional recurrent neural network,” in *Proc. ICASSP*, 2017, pp. 771–775.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Proc. NIPS*, 2017, pp. 5998–6008.

[16] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and Kazuya Takeda, “Weakly-supervised sound event detection with self-attention,” in *Proc. ICASSP*. IEEE, 2020, pp. 66–70.

[17] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D Plumbley, “Sound event detection of weakly labelled data with CNN-transformer and automatic threshold optimization,” *IEEE TASLP*, vol. 28, pp. 2450–2460, 2020.

[18] Niko Moritz, Gordon Wichern, Takaaki Hori, and Jonathan Le Roux, “All-in-one transformer: Unifying speech recognition, audio tagging, and event detection,” *Proc. Interspeech 2020*, pp. 3112–3116, 2020.

[19] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and Kazuya Takeda, “Conformer-based sound event detection with semi-supervised learning and data augmentation,” in *Proc. DCASE2020 Workshop*. DCASE, 2020.

[20] Gangyi Tian, Yuxin Huang, Zhirong Ye, Shuo Ma, Xiangdong Wang, Hong Liu, Yueliang Qian, Rui Tao, Long Yan, Kazushige Ouchi, and Reinhold Ebbers, Janek Haeb-Umbach, “Sound event detection using metric learning and focal loss for DCASE 2021 task 4,” Tech. Rep., DCASE2021 Challenge, June 2021.

[21] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma, “Linformer: Self-attention with linear complexity,” *arXiv preprint arXiv:2006.04768*, 2020.

[22] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya, “Reformer: The efficient transformer,” *arXiv preprint arXiv:2001.04451*, 2020.

[23] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun, “Explicit sparse transformer: Concentrated attention through explicit selection,” *arXiv preprint arXiv:1912.11637*, 2019.

[24] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret, “Fast transformers with clustered attention,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 21665–21674, Curran Associates, Inc.

[25] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind, “An attention free transformer,” *arXiv preprint arXiv:2105.14103*, 2021.

[26] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon, “Fnet: Mixing tokens with fourier transforms,” *arXiv preprint arXiv:2105.03824*, 2021.

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

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016, pp. 770–778.

[29] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret, “Transformers are RNNs: Fast autoregressive transformers with linear attention,” in *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh, Eds. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165, PMLR.

[30] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon, “Sound event detection in domestic environments with weakly labeled data and soundscape synthesis,” in *Proc. DCASE2019 Workshop*, New York City, United States, October 2019.

[31] Scott Wisdom, Hakan Erdogan, Daniel Ellis, Romain Serizel, Nicolas Turpault, Eduardo Fonseca, Justin Salamon, Prem Seetharaman, and John Hershey, “What’s all the fuss about free universal sound separation data?,” in *arXiv preprint arXiv:2011.00803*, 2020.

[32] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, “Scaper: A library for soundscape synthesis and augmentation,” in *Proc WASPAA*. IEEE, 2017, pp. 344–348.

[33] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon, “Sound event detection in domestic environments with weakly labeled data and soundscape synthesis,” Proc. DCASE2019 Workshop, June 2019.

[34] Antti Tarvainen and Harri Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[35] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.