symmetry

MDPI

Article

Adaptive Memory-Controlled Self-Attention for Polyphonic Sound Event Detection

Mei Wang [1], Yu Yao [2,3], Hongbin Qiu [2,3] and Xiyu Song [2,3,*]

[1] School of Information Science & Engineering, Guilin University of Technology, Guilin 541006, China; mwang@glut.edu.cn
[2] Ministry of Education Key Laboratory of Cognitive Radio and Information Processing, Guilin 541006, China; berskiijenkins@gmail.com (Y.Y.); qiuhb@guet.edu.cn (H.Q.)
[3] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541006, China
* Correspondence: songxiyu@guet.edu.cn

Abstract: Polyphonic sound event detection (SED) is the task of detecting the time stamps and the class of sound event that occurred during a recording. Real life sound events overlap in recordings, and their durations vary dramatically, making them even harder to recognize. In this paper, we propose Convolutional Recurrent Neural Networks (CRNNs) to extract hidden state feature representations; then, a self-attention mechanism using a symmetric score function is introduced to memorize long-range dependencies of features that the CRNNs extract. Furthermore, we propose to use memory-controlled self-attention to explicitly compute the relations between time steps in audio representation embedding. Then, we propose a strategy for adaptive memory-controlled self-attention mechanisms. Moreover, we applied semi-supervised learning, namely, mean teacher–student methods, to exploit unlabeled audio data. The proposed methods all performed well in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Sound Event Detection in Real Life Audio (task3) test and the DCASE 2021 Sound Event Detection and Separation in Domestic Environments (task4) test. In DCASE 2017 task3, our model surpassed the challenge's winning system's F1-score by 6.8%. We show that the proposed adaptive memory-controlled model reached the same performance level as a fixed attention width model. Experimental results indicate that the proposed attention mechanism is able to improve sound event detection. In DCASE 2021 task4, we investigated various pooling strategies in two scenarios. In addition, we found that in weakly labeled semi-supervised sound event detection, building an attention layer on top of the CRNN is needless repetition. This conclusion could be applied to other multi-instance learning problems.

Keywords: sound event detection; self-attention mechanism; convolutional recurrent neural networks; adaptive memory control; semi-supervised learning

Citation: Wang, M.; Yao, Y.; Qiu, H.; Song, X. Adaptive Memory-Controlled Self-Attention for Polyphonic Sound Event Detection. *Symmetry* **2022**, *14*, 366. https://doi.org/10.3390/sym14020366

Academic Editor: Gianluca Vinti

Received: 20 January 2022
Accepted: 8 February 2022
Published: 12 February 2022

1. Introduction

Driven by the annual Detection and Classification of Acoustic Scenes and Events (DCASE) challenges [1–3], sound event detection (SED) is gaining interest from the industry [4,5]. Differently from the typical classification problem that assigns an audio example to one or more classes, e.g., sound scene classification [6], SED not only requires detection of the time stamps, but also the class(es) of sound event in a recording. Sound events do not often occur in isolation, but tend to overlap with each other. Recognizing such overlapping sound events is referred to as polyphonic sound event detection [7]. Polyphonic SED is always considered as a multi-label multi-class problem. Consequently, approaches based on machine learning have shown to be especially effective [8,9]. Some deep learning models, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and their combination model, CRNN [10], have recently prevailed and have shown improved performances in SED. However, based on the fixed sizes of their convolution filters, CNNs make decisions using only limited context. RNNs solve context

limitations, but take no notice of the fact that some frames in context are more important than others when making decisions. With the great success of the transformer [11], the attention mechanism has received considerable attention, as it can address the above worry about RNNs.

The self-attention mechanism, also called intra-attention, is one of the extensions of the attention mechanism. It models relations within a single sequence. Each embedding in one time step is a weight sum representation of all of the rest of the time steps within the sequence. Self-attention has been used successfully in a variety of tasks, including reading comprehension, abstractive summarization, textual entailment [12], and automatic speech recognition [13]. Many methods in the natural language processing (NLP) field, such as RNNs, connectionist temporal classification (CTC) [14], and dynamic time warping (DTW), have been widely and successfully used for references in sound event detection [15]. Thus, the attention mechanism has been introduced into SED; for example, a self-attention model was deployed to help an SED model to distinguish between relevant and irrelevant parts of a weakly labeled audio clip [16]. Kong et al. adopted an attention model in audio classification and explained it from a probability perspective [17]. Wang et al. applied self-attention mechanisms based on transformer attention [18]. However, none of these studies have analyzed how many neighbors around each frames in embedding should be considered representative. In addition, when considering NLP and SED, it should be noted that the recurrence relations in recordings are not always certain, and environmental sounds lack inherent structure. Consequently, SED tasks cannot always improve performance by including more neighbors around each frame as machine translation language modeling task did in [19]. In [20], Dai et al. developed a self-attention based Transformer-XL which can learn much longer dependency than vanilla transformer [11]. This method improved the state-of-the-art results in language modeling. In text data, there are certain relations between phonemes in a word and consecutive word relations in a sentence. In view of the uncertainty of long term dependency in sound events, a natural idea is to do exactly the opposite of getting longer dependency—that is, to constrain the attention dependency length. Pankajakshan et al. first implemented this idea in sequential attention mechanisms [21]. The attention function in sequential attention can be described as mapping a *query* and a set of *key–value* pairs to an output [11]. By selecting different lengths of *key–value* set, they changed the attention width and controlled the memory used for attention. However, Pandajakshan et al. worked with a fixed attention width, and there is no general agreement about how long the attention width should be. These results were limited to the development datasets, and are therefore not representative of attention width in general. This indicates a need for automatic memory-controlled attention in SED.

To obtain a model with better recognition performance, a large amount of training data is greatly needed. Training data should contain audio and corresponding annotations, e.g., onsets, ends, and categories. The annotator should listen carefully to assign timestamps to a sound event; he may repeat the audio a couple of times to be sure when sounds overlap. Collecting training data is therefore a difficult task. To relieve the data annotation difficulty, Wang et al. simplified the annotation process [22,23]. Their work can be considered a trade-off between accuracy and annotation cost. Another example is weakly-supervised SED [24]. In weakly-supervised SED, the training audio data only contain sound event tags without time boundaries. This method releases the audio annotation burden to some extent. However, there is nearly infinite potential audio training data on social media servers. Leveraging this unlabeled audio data to improve SED is a trend.

In this study, we extended our previous work on DCASE 2021 task4 [25]. We adopted a mean teacher method [26] to develop a semi-supervised learning SED system which leverages both the labeled data (strongly labeled, weak labeled) and the unlabeled data. A self-attention mechanism was introduced to memorize dependencies of features extracted from the CRNN. Next, by constraining the self-attention function to a certain length of compact neighborhood relative to each frame, we evaluated the potential of memory-controlled sequential self-attention in DCASE 2017 and DCASE 2021. We propose an

adaptive memory-controlled self-attention mechanism that can learn optimal attention width. The experimental results show that the proposed method improves the detection performance, especially in DCASE 2021 task4, scenario 2; and DCASE 2017 task3. Finally, we found that pooling is vital for sound event detection. We evaluated all the pooling strategies with polyphonic sound detection score (PSDS) metrics [27]. In a nutshell, our contributions are the following:

- A supervised memory-controlled attention model that improves sound event detection by a large margin, without manually optimizing towards dataset-specific attention width;
- Using the SOTA evaluation criterion, we verified our approach with two publicly available datasets;
- We studied the weakly-supervised SED pooling strategy on DCASE 2021 task4. This result can be a reference for pooling selection.

The paper is structured as follows: Section 2 describes the detection system from the top level. In Section 3, we propose the memory-controlled sequential self-attention model (under supervised and semi-supervised learning). We show our experiments and comparative data in Section 4. We discuss our findings in Section 5.

2. Related Work

2.1. Basic Systems Structures

The basic SED systems are in Figure 1. There are two stages in an SED system's development—namely, the learning stage and usage stage. Supervised SED model means all the data involved in the learning stage are annotated, e.g., *at 5.1–8 s, alarm bell ringing*. In the supervised learning stage, annotations are used as reference information to automatically learn a mapping between audio and class labels. The mel frequency cepstral coefficients (MFCCs) and log-mel spectrogram are the most commonly used time–frequency feature representations of a raw audio waveform. Take feature representation $O_n \in \mathbb{R}^F$ extracted from $t = 1, 2, 3, \ldots, N$, where F is the number of features, and take encoded annotation $Y_n \in \mathbb{R}^C$, where C is the predefined number of sound event classes. If, according to the reference annotations, the class with the Cth label is present in analysis frame n, then $Y_{c,n}$ are set to 1 and 0. In Figure 1, the colored parts of the label represent the value 1, and blank represents 0. In the usage stage, the learned model takes audio features that experience the same processing block and feature extraction as input, and then outputs the two-dimensional prediction matrix. Human-readable annotations could be obtained by decoding this matrix.

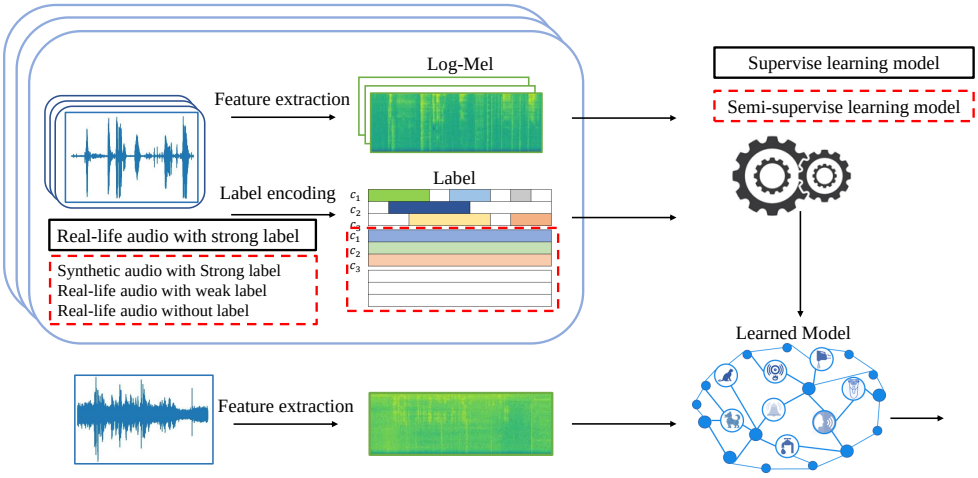

Figure 1. Supervised and semi-supervised SED systems. Black rectangles and red dashed rectangles represent supervised and semi-supervised training material, respectively.

2.2. Weakly-Supervised Sound Event Detection

Weakly-supervised SED leverages weakly labeled data. The key to this is to aggregate the outputs of neural networks for a tag predictor. This aggregation operation is also referred to as pooling. which summarizes frame-level probabilities into clip-wise probabilities. McFee [28] analyzed the commonly seen temporal mean and max-pooling. Their results on DCASE 2017 suggest that temporal max-pooling performs well when events are short. Wang [29] compared five pooling functions on DCASE 2017 and showed linear softmax pooling to be the best among the five. Lin et al. [30] mainly focused on attention pooling. After combining it with the proposed specialized decision surface method, they achieved impressive results on DCASE 2018 task4 and DCASE 2019 task4. Lin's study also demonstrated that the embedding-level pooling approach tends to outperform the instance-level one. This conclusion agrees with Kong's research on audio tagging tasks [31], which achieved a mean average precision (mAP) of 0.369 on audioSet tagging with feature-level attention pooling. In spite of the different names the latter papers used, their models are similar under the hood. If the bottleneck layer's output (where the aggregation occurs) has a fixed number of sound classes, it will be instance-level pooling in Lin's model and decision-level pooling in Kong's; otherwise, embedding-level in Lin's and feature-level in Kong's. The unfixed version will likely pull more information from feature space to aggregate, leading to a better result. Heinrich Dinkel [32] proposed a new post-processing method utilizing both frame-level and clip-level output, which can be considered SOTA in terms of tagging in DCASE 2018. All this work only studied the pooling on weakly-supervised models, evaluated by either a collar-based or a segment-based criterion. In our work, we investigate pooling strategies in semi-supervised learning. Moreover, we provide insights into pooling in a specific scenario, which can supplement the aforementioned research.

3. Methods

In both supervised and semi-supervised learning models, we implemented the proposed adaptive memory-controlled sequential self-attention on top of a CRNN model, as illustrated in Figure 2.

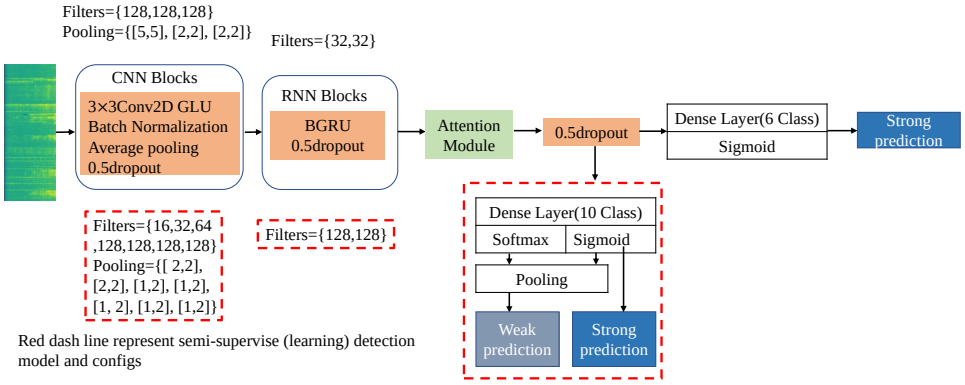

Figure 2. The proposed supervised and semi-supervised models are in the upper and lower parts, respectively. The different configurations of the two are indicated by the red dashed line.

3.1. Model Overview

The CRNN models used in the two systems are basically the same, except for the output layer and the number of layers. The strongly labeled audio in DCASE 2017 were difficult to collect, and thus there are fewer data compared to DCASE 2021. Consequently, we reduced the CNN layers to three to avoid overfitting. The output layers have six sigmoid units, which correspond to the six classes of sound events. The supervised leaning model for DCASE 2017 was trained by minimizing binary cross entropy. Figure 3 illustrates the mean teacher method and the semi-supervised learning model for DCASE 2021. The mean teacher method averages model weights instead of performing label prediction. It achieved state-of-the-art results without changing the network architecture. Moreover,

any established model can be built under this method. There are two models, namely, the student model and the teacher model, in the mean teacher–student method. The student model and the teacher model share the same CRNN with adaptive memory-controlled architecture, but the teacher model's weights are updated as exponential moving averages of the student model's weights. The whole semi-supervised learning model was trained by using three kinds of datasets: strongly labeled, weakly labeled, and unlabeled audio data. The student model was trained by strongly labeled data and weakly labeled data; the loss function of the student model includes strong label loss and weak label loss; we used binary cross entropy (BCE) to compute this supervised loss. Note that the unlabeled data were passed forward to the student model so that there were weak and strong predictions of unlabeled data. We did not use the data for student loss computing, but for distribution consistency loss, which is also called self supervised loss. Mean square error (MSE) loss was used to compute self supervised loss between predictions from student and teacher models. Finally, the total semi-supervised model was trained by minimizing the sum of supervised loss and self supervised loss. Inherently, self supervised loss obliged the predictions from both teacher and student models to be as likely as possible, and thus exploited large amounts of unlabeled audio data. Both trained models can be used for prediction, but the teacher's predictions are more likely to be right [26].

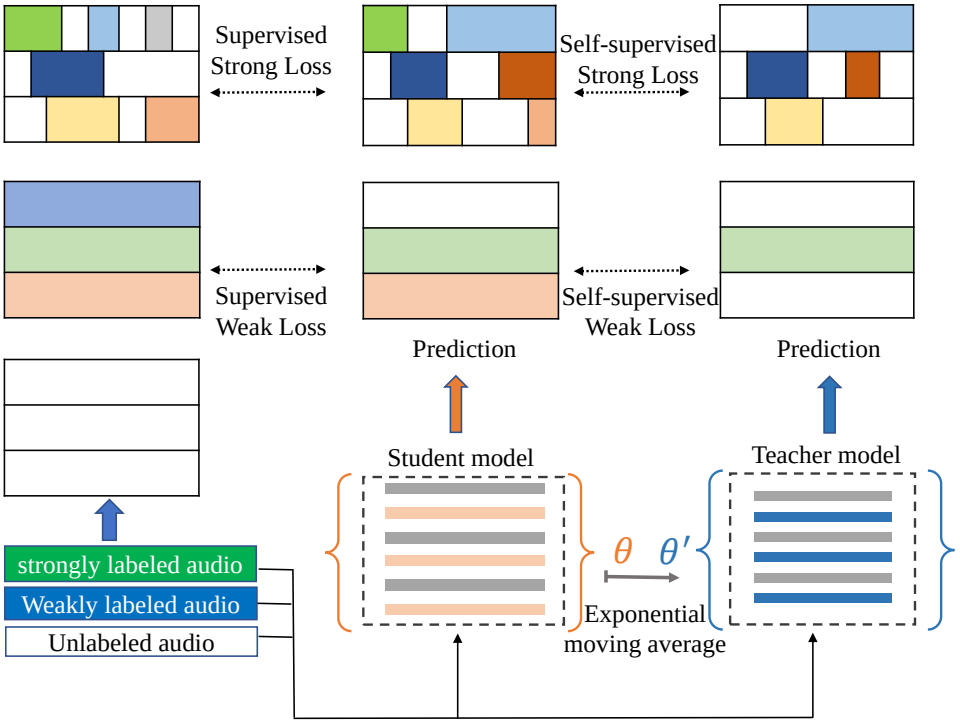

Figure 3. Mean teacher method. The colored rectangles represent the occurrences of a sound event in the label. The length of rectangles with different colors are the timestamps of an occurred sound event.

3.2. Pooling Function

Pooling in SED means to aggregate a neural network's (student model and teacher model in Figure 3) frame-level probability outputs to recording-level predictions. Note that this enables a number of dimension reductions, which is a differentiating characteristic from pooling after convolution. Let the probability of an event class at the n-th frame be $y_n \in [0, 1]$, and the aggregated along the time axis clip-level probability be $y \in [0, 1]$. We introduce max-pooling [33], average pooling [34], linear softmax pooling [35], exponential softmax pooling [29], attention pooling [30,31], and auto-pooling [28].

The max-pooling function takes the largest class presence probability y_n at the n-th frame as y. This can be formulated as:

$$y = \max_n(y_n) \tag{1}$$

This max-pooling strategy is not able to consider that some sound classes are less likely than the maximum probability, yet occurred multiple times. All the other pooling functions are designed to improve on it by assigning different weights. Average pooling assigns equal weights for every prediction in one frame:

$$y = \frac{1}{N}\sum_n y_n \tag{2}$$

where N is the total number of frames in a recording. The linear and exponential softmax pooling assign weights by the output probabilities themselves, as in the following two equations:

$$y = \frac{\sum_n y_n^2}{\sum_n y_n} \tag{3}$$

$$y = \frac{\sum_n y_n \; exp(y_n)}{\sum_n exp(y_n)} \tag{4}$$

These two pooling strategies tend to be focused on positive prediction. Attention pooling assigns weights from a learnable layer in the network. Attention pooling was used as the DCASE 2021 task4 baseline [3]. Let ω be the learnable parameters. We have the clip-level probability:

$$y = \frac{\sum_n y_n \; \omega_n}{\sum \omega_n} \tag{5}$$

Auto pooling is an improved version of exponential softmax pooling. A trainable parameter α is used as a scalar:

$$y = \frac{\sum_n y_n \; exp(\alpha \; y_n)}{\sum_n exp(\alpha \; y_n)} \tag{6}$$

When $\alpha = 0$, Equation (4) reduces to unweighted average pooling; when $\alpha = 1$, Equation (4) simplifies to exponential softmax pooling; and when $\alpha \to \infty$, Equation (1) approaches max-pooling. This can be assumed to automatically adapt to and interpolate between different pooling behaviors [28].

3.3. Semi-Supervised Model

In the student model, we use a CRNN as the hidden state feature representation extractor, which takes a log-mel spectrogram as input. The CRNN architecture is as follows. The CNN part consists of seven convolution layers. The filter size at each layer increases at a power of 2. The first layer has 16 filters, the second 32, and the third 64. The remaining four layers have 128 filters in total. Batch normalization and max-pooling are performed after every layer of CNN along the frequency axis. In order to introduce non-linear characteristics to a CRNN network, a learnable gated activation function called gated linear units (GLU) is used instead of using sigmoid or ReLU activation. Dropout is used as a regularizer after every layer of CNN. Then the resulting feature maps are fed as input to two bi-directional GRUs [36] with 128 RNN cells. After that, the extracted hidden state representations are processed by proposed adaptive memory-controlled sequential self-attention to derive improved hidden state representations. A layer of a time-distributed fully-connected network is followed by the final output layer with 10 sigmoid units as the number of sound event class labels in the dataset. Finally, we obtain the sequence of class classification probabilities in the student model with network parameter θ : $Y_{student} = (\hat{y}_{tag}, \hat{y}_n|\theta)$. It contains both clip-level predictions $\hat{y}_{tag}$ and frame-level predictions $\hat{y}_n$), where n is the

index of frames. $n \in 1, \ldots N(\hat{y}_{tag}$ is obtained by pooling, which reduces $\hat{y}_n)$ along the time axis. The supervised loss is computed as follows:

$$Loss_{supervised} = Loss_{student} = Loss_{weak} + Loss_{strong}, \tag{7}$$

where:

$$Loss_{weak} = \sum_{i=1}^{C} BCELoss(y_{tag}, \hat{y}_{tag}), \tag{8}$$

$$Loss_{strong} = \frac{1}{n} \sum_{n=1}^{N} \sum_{i=1}^{C} BCELoss(y_n, \hat{y}_n), \tag{9}$$

y_{tag} and y_n are the ground truth of the clip-level label and frame-level label, respectively. The teacher model shares the same CRNN with the adaptive memory-controlled network architecture; however, after the weights of the student model have been updated with gradient descent, the teacher model weights are updated as exponential moving averages of the student model weights [26]. In this paper, we use θ' to indicate the weights of teacher model and θ to indicate the weights of the student model. We define θ'_t at training step t as the exponential moving average of the θ weight:

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t, \tag{10}$$

where α is a smoothing coefficient hyperparameter. The sequence of class classification probabilities of teacher model $Y_{teacher} = (\hat{y}_{tag}, \hat{y}_n|\theta)$ also contains both clip-level label predictions $\hat{y}_{tag}$ and frame-level label predictions $\hat{y}_n$. The self supervised loss is the MSE loss between the inference from teacher model with θ' and the student model with θ:

$$Loss_{self\_supervised} = Loss_{self\_weak} + Loss_{self\_strong}, \tag{11}$$

$$Loss_{self\_weak} = \sum_{i=1}^{C} MSELoss((\hat{y}_{tag}|\theta'), (\hat{y}_{tag}|\theta)), \tag{12}$$

$$Loss_{self\_strong} = \sum_{i=1}^{C} MSELoss((\hat{y}|\theta'), (\hat{y}|\theta)), \tag{13}$$

Finally, the total semi-supervised model is trained by minimizing the sum of supervised loss and self supervised loss as follows:

$$Loss_{total} = Loss_{supervised} + Loss_{self\_supervised}, \tag{14}$$

3.4. Self-Attention Mechanism

Key, value, and query are commonly used terms to describe attention mechanism. Given a query vector, the weight coefficient of the value corresponding to each key is obtained by calculating a similarity score in key–value pairs. Then the final attention value is weighted and summed. This can be formulated as:

$$Attention(Q,K,V)=softmax(Q,K)V, \tag{15}$$

where Q, K, V are the query, key, and value matrices, respectively. When $Q = K = V$, it is defined as "self-attention."

3.5. Adaptive Memory-Controlled Self-Attention

In this paper, we use $H = (h_1, h_2, \ldots, h_N)$ to represent the hidden state representations that the CRNN extracts. where N denotes the total number of frames in a recording. In terms of query, key, and value representations, $Q = K = V = h_n$. To explicitly compute

the relations between time steps, we apply memory-controlled self-attention on H. The resulting improved representations $\tilde{H} = (\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_N)$, where

$$\tilde{h}_n = \sum_{i=n-\frac{L}{2}}^{n+\frac{L}{2}} \alpha_i^n h_i; n \in \{1, \ldots, N\}. \tag{16}$$

α_i^n is the attention weight value computed using a similarity function:

$$\alpha_i^n = softmax(s_i^n), \tag{17}$$

$$s_i^n = score(h_n, h_i); i, n \in \{1, \ldots, N\}. \tag{18}$$

Additive score functions instead of scale-dot score function in [11] are used as follows:

$$score(h_n, h_i) = V_\alpha^\top tanh(W_\alpha[h_y; h_i]), \tag{19}$$

where V_α and W_α are the weight terms of the score functions and $\top$ denotes transposition. On tensor level, $score(h_n, h_i)$ is symmetric matrices. We constrain the self-attention function as in Equation (16) so that the improved representation at each time step is computed only on its nearest L neighbors, namely, attention width. That is:

$$Q = h_n, \ and \ K_L = V_L = h_{n-\frac{L}{2},\ldots,h_n,\ldots h_{n+\frac{L}{2}}}, \tag{20}$$

On a tensor level, this turns the symmetric score matrices into band matrices.

The vanilla self-attention in [11] computes representation in each time step $\tilde{h}_N$ with respect to all the other time steps. Inherently, sound events occurring in 10 s recordings typically lack the syntactic and semantic relations that human language has [37]; that is to say, frame-level representation h_n among sound events could lead to incorrect $\tilde{h}_N$, which would do no good to the detection model. An insight about Equation (16) is that it just takes the structured part within a sound event into consideration. Since there are no general rules for choosing attention width, and in order to overcome the limitation of a fixed attention width, we propose adaptively controlling the attention width. When we compute the similarity score in Equation (18) we add a masking function to control the width of the attention. A masking function is a non-increasing function that maps a distance among frames to a value in $[0, 1]$. This idea is inspired by the attention span in machine translation in [19]. A difference between language model tasks and SED is that the attention mechanism is applied to word level embedding in the language model and frame-level embedding in SED. With the mask function, the attention scores among frames are weighted with their distances. We take the following soft masking function m_z parameterized by a real value z in $[1, N]$:

$$m_z(x) = min\left[\max\left[\frac{1}{R}(R + z - x), 0\right], 1\right], \tag{21}$$

where R is a hyper-parameter that controls its softness. $min[\cdot], max[\cdot]$ are the operators used to get the minimum and maximum of the two values. The attention weights from Equation (17) are then computed as:

$$\alpha_i^n = m_z(n-i)softmax(s_i^n) = \frac{m_z(n-i)exp(s_i^n)}{\sum_{i=1}^N m_z(n-i)exp(s_i^n)}, \tag{22}$$

where $exp(\cdot)$ is the exponentiation operator.

4. Experiment

In this section, we introduce the DCASE challenge and datasets, with which we performed a series of experiments to evaluate the memory-controlled model and pooling methods. All the training configuration information is also in this section.

4.1. DCASE 2017 Task3

DCASE 2017 task3 requires evaluating the performances of sound event detection systems in multi-source conditions similar to our everyday life.

4.2. DCASE 2021 Task4

The goal of DCASE 2021 task4 [8] is to evaluate systems for the detection of sound events. This task is the follow-up to DCASE 2020 task4 [3]. The challenge consists of detecting sound events within audio clips using training data from real recordings, both weakly labeled and unlabeled, and synthetic audio clips that are strongly labeled. This task requires evaluating systems with two scenarios by specifying corresponding PSDS parameters. Scenario 1 focuses on the localization of the sound event: the system in scenario 1 needs to react fast upon an event detection (to trigger an alarm, adapt home automation system, etc.). Scenario 2 puts more emphasis on the system's ability to distinguish among events, and the reaction speed requirement is less rigid.

4.3. Evaluation Criterion

In this study, we used polyphonic SED metrics proposed in [38] and state-of-art polyphonic sound detection score (PSDS) metrics described in [27] for evaluation.

(1) Polyphonic SED metrics. There are two ways of comparing system outputs and ground truth with polyphonic SED metrics: segment-based and event-based. Segment-based metrics compare system outputs and ground truth in short time segments, e.g., 1 s. Event-based metrics compare system outputs and ground truth event by event. Tolerance is allowed, e.g., a $\pm$100 ms collar. In this paper, we use segment-based F-score, event-based F-score, and segment-wise error rate. F-score F is calculated as follows:

$$p = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F = \frac{2TP}{2TP + FP + FN}, \tag{23}$$

Note that the segment-based and event-based Precision P, Recall R, and F-score F are calculated based on corresponding intermediate statistics. TP is the number of True Positive, FP is the number of false positives, and FN is the number of false negatives. The error rate ER measures the amount of errors in terms of insertions (I), deletions (D), substitutions (S) and the number of active sound event N in segment k.:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)}, \tag{24}$$

(2) PSDS metrics. PSDS metrics allow for system comparisons independently from operating points (OPs) and provide more global insights into system performance. Furthermore, by specifying a system's parameters, it can be tuned to varying applications in order to match a variety of user experience requirements. As in [27], we used predefined detection tolerance criterion (DTC) ρ_{DTC}, ground truth tolerance criterion (GTC) ρ_{GTC}, and the cross-trigger tolerance criterion (CTTC) ρ_{CTTC}. We then computed the TP ratio $r^*_{TP,c}$, FP ratio $R^*_{FP,c}$, the CR rate $R^*_{CT,c,\hat{c}}$, then we have the effective FP rate (eFPR):

$$eFPR : e^*_c \triangleq R^*_{FP,c} + \alpha_{CT} \frac{1}{|C| - 1} \sum_{\hat{c} \in C, \hat{c} \neq c} R^*_{CT,c,\hat{c}}, \tag{25}$$

In Equation (25), α_{CT} is a weighting parameter, $c \in C$ is the sound event class, and $|C|$ is the total number of classes.

$$\mu_{TP} = \frac{1}{C} \sum_{c \in C} r_{TP,c}, \tag{26}$$

$$\sigma_{TP} = \sqrt{\frac{1}{C} \sum_{c \in C} (r_{TP,c} - \mu_{TP})^2}, \tag{27}$$

Using both the exception (Equation (26)) and standard deviation (Equation (27)) of TP ratios across classes, the effective TP ratio (eTPR) is computed as eTPR: $re \triangleq \mu_{TP} - \alpha_{ST} * \sigma_{TP}$, where α_{ST} adjusts the cost of instability across classes. The coordinates of $(eFPR, eTPR)$ form the PSD ROC curve. The normalized area under the PSD curve is the PSD score (PSDS). Given dataset ground truth Y, and the set of evaluation parameters, $(\rho_{DTC}, \rho_{GTC}, \rho_{CTTC}, \alpha_{CT}, \alpha_{ST})$, a PSDS of a system detection is:

$$PSDS \triangleq \frac{1}{e_{max}} \int_0^{e_{max}} r(e) de, \tag{28}$$

where e_{max} is the maximum eFPR value of interest for the SED application under evaluation.

In DCASE 2021 task4, the system detection will be evaluated in two scenarios that emphasize different system properties as follows. In scenario 1, the system needs to react fast upon event detection (e.g., to trigger an alarm or adapt a home automation system). The localization of the sound event is then very important. The PSDS parameters reflecting these needs are:

- Detection tolerance criterion ρ_{DTC}: 0.7;
- Ground truth intersection criterion ρ_{GTC}: 0.7;
- Cross-trigger tolerance criterion ρ_{CTTC}: 0.3;
- Cost of CTs on user experience α_{CT}: 0;
- Cost of instability across class α_{ST}: 1.

In scenario 2, the system must avoid confusing between classes, but the reaction time is less crucial than in the first scenario. The PSDS parameters reflecting these needs are:

- Detection tolerance criterion ρ_{DTC}: 0.1;
- Ground truth intersection. Criterion ρ_{GTC}: 0.1;
- Cross-trigger tolerance criterion ρ_{CTTC}: 0.3;
- Cost of CTs on user experience α_{CT}: 0.5;
- Cost of instability across class α_{ST}: 1.

4.4. Implementation Details

All recordings from the development dataset were resampled to 16 kHz and down-sampled to mono. Then we extracted a 128-dimensional log-mel spectrogram using a short-time Fourier transform (STFT) with a 2048 FFT window, a hop length of 256, and a sample rate of 16 kHz. A temporal subsampling rate of 4 was used on the resulting log-mel spectrogram. In DCASE 2017 we used a batch size of 32 to train for 30 epochs. Four-fold cross-validation was used to obtain the best model. For DCASE 2021, the median filter was used to smooth the predictions. We trained the network for 200 epochs using a binary cross-entropy loss function with a learning rate of 0.001; we adopted an exponential warm up for the first 50 epochs. We applied a batch size of 48 (1/4 synthetic data, 1/4 weak-label data, 1/2 unlabeled data). Mixup [39] with a rate of 0.5 and FilterAugment [40] were used to augment audio data.

5. Discussion

In this section, we describe a series of experiments with the memory-controlled self-attention mechanism in supervised and semi-supervised models. For the semi-supervised

model we also investigated the behavior of six pooling methods in DCASE 2021 task4. We also make a comparison to other solutions (algorithms) provided by other researchers in this section.

Experimental Results and Analysis

In DCASE 2017, for the purpose of making contrastive analysis of adaptive memory-controlled self-attention, we trained the supervised learning model via real-life strong label audio data as the challenge requires. We first heuristically chose a set of fixed length attention widths for both the DCASE 2017 supervised model and the DCASE 2021 semi-supervised model. Then we analyzed the corresponding SED performances. Lastly, we tested our adaptive memory-controlled sequential attention model. We show the proposed model's performance on DCASE 2017 in Table 1. Since we used four-fold cross-validation, all the results from the table are the averages of four attempts. These results suggest that the improved features of the self-attention mechanism do help in real-life, long duration sound event detection. Note that it has a better error-rate and F1-score than the winning system in DCASE 2017 [41]. This might be owed to the fact that sound events with a long duration are common in real-life recordings; for instance, a *car* produces a sound lasting 105.8s on average in each recording. The attention memory-controlled model takes the appropriate amount of context of the sequence into account. For a certain length of context, the embedding in each frame is the weight sum of the other; some frames' features are focused on more, which enables the model to capture the differentiating features among sound events. From the experimental results, we found that it cannot always gain improved performance by getting longer attention width. With an attention width of 300, for example, the performance is inferior to having an attention width of 200. By using an attention width of 200, we get the best performance. Our proposed adaptive model reached the same level of performance as the fixed length model.

Table 1. Experimental results of DCASE 2017. The baseline [42] system is using CRNN without any memory controlled. The bold font indicates the best system.

Detection System	Segment-Based	
	Error Rate	F1-Score
Baseline [42]	0.9358	42.8%
Winner system [41]	0.7914	41.7%
Attention width 20	0.711	48.3%
Attention width 50	0.7063	48.1%
Attention width 100	0.7065	47.4%
Attention width 200	**0.6810**	**49.6%**
Attention width 300	0.6980	49.5%
Adaptive strategy	0.6927	49.6%

In DCASE2021, Zheng uses Selective kernel to design CNN and achieve the first rank [43]. Wang's CNN-transformer [44] also leverages attention mechanism, however, they focus on attention among channels. Various augmentation, post-processing and ensemble learning are used in Wang's work. Dinkel [45] works on fewer parameters, which enables a lightweight model and is more adaptable to practical application. We show their PSDS evaluation result in Table 2. However, note that although the PSDS1 and PSDS2 are from one work, the detection system's configuration (e.g., model architecture, post-processing techniques) might be different. For our semi-supervised model, we first show the pooling strategy's influence without any memory control in Table 2. Then we present our memory-controlled model.

Table 2. Experimental results in DCASE 2021. PSDS1 means polyphonic sound event detection score in scenario 1. PSDS2 means polyphonic sound event detection score in scenario 2. The third column is the sum of PSDS1 and PSDS2, which is the DCASE challenge ranking criterion. attn20 is an abbreviation of a attention width of 20. Note that F-score and error rate are computed by macro average. The memory-controlled model with a set of memory widths and baseline [3] used attention pooling. The best in the last row of the table is the sum of attention pooling PSDS1 and auto pooling PSDS2.

System				Event-Based		Segment-Based	
	PSDS1	PSDS2	Total	F-Score	Error Rate	F-Score	Error Rate
max pooling [33]	0.37	0.65	1.02	42.35%	1.08	73.89%	0.46
avg pooling [34]	0.06	0.74	0.80	18.71 %	1.28	67.32 %	0.83
lin pooling [35]	0.38	0.65	1.03	48.37 %	0.94	76.88	0.45
exp pooling [29]	0.07	0.74	0.81	19.77 %	1.27	67.84 %	0.81
attn pooling [31,46]	0.39	0.60	0.99	46.08 %	1.04	75.44 %	0.46
auto pooling [28]	0.08	0.74	0.82	20.48 %	1.26	68.11 %	0.79
attn20	0.14	0.65	0.79	19.87%	1.09	63.12%	0.58
attn50	0.18	0.65	0.83	23.06%	1.06	65.29%	0.55
attn100	0.17	0.62	0.79	24.79%	1.05	64.66%	0.56
attn200	0.05	0.73	0.74	19.51%	1.24	66.70%	0.85
Baseline [3]	0.34	0.52	0.86				
CNN-transformer [44]	0.37	0.72	1.09				
winner [43]	0.45	0.74	1.19				
light weight [45]	0.37	0.60	0.97				
best	0.39	0.74	1.13				

The evaluation results of Equation (28) in DCSAE2021 are in Figures 4 and 5. We can observe the trade-off behavior of PSDS scenario 1 and PSDS scenario 2. Among the six pooling strategies, linear softmax pooling performed the best. Max-pooling performed nearly equally to linear softmax pooling in PSDS, but linear softmax showed better localization ability. With linear softmax, a larger prediction will be boosted and the frame-level probabilities are driven to extremes 0 and 1; it enables the model to perform "bold" predictions. Average, exponential, and auto pooling achieved 0.74 PSDS scores in scenario 2 and had inferior performance in scenario 1. Many false positive frames were produced, causing a higher error rate, as shown in Table 2. With attention pooling, the contribution (the frame-level probability will be boosted or suppressed) of each frame is generated by the learned attention weight, which is referred to as the loss function. A problem may occur when a recording-level probability is indeed small, but a larger weight was assigned [29]. As a result, the model cannot distinguish between classes. Although attention pooling in PSDS1 slightly outperformed the linear softmax pooling, it does not seem to present a good deal because of the cost of the extra learnable weights.

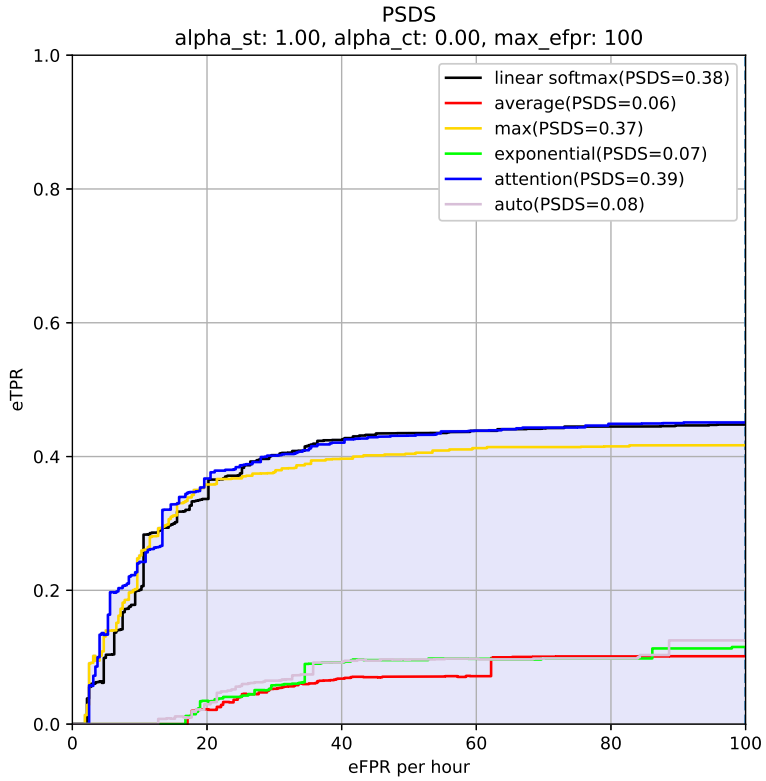

Figure 4. PSDS evaluation result of six pooling strategies in scenario 1. PSDS is the normalized area under the PSD curve. In this scenario, we focus on investigating the temporal localization ability.

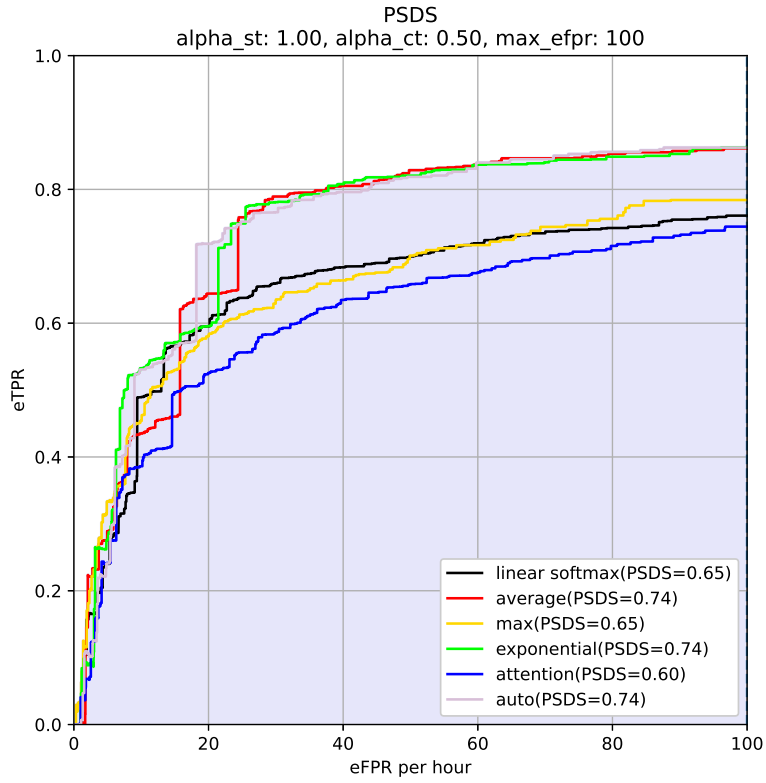

Figure 5. PSDS evaluation result of six pooling strategies in scenario 2. PSDS in scenario 2 reflect the classification ability among inter-class of the detection system.

Compared to the baseline, the proposed memory-controlled self-attention improved the PSDS in scenario 2. However, the PSDS and event-based F1 scores in scenario 1 were heavily reduced. This indicates that a reaction-time-rigid system does not benefit from an attention mechanism; on the contrary, an attention mechanism results in inappropriate detection of timestamps of sound events. Due to the event-based F1 scores being computed by comparing the ground truth labels to annotations event by event, the detected sound events tended to surpass the predefined collar, leading to a rapid decrease in event-based F1 score compared to segment-based F1 score. In Table 2, we can also see the trade-off between different lengths of attention width. In Figure 1, the attention layer is built on top of the CRNN output layer. The self-attention mechanism can be regarded as assigning a weight to the output tensor of the CRNN layer. Taken together, we consider that adding an attention layer to CRNN is a needless repetition in a model that uses both recording-level and frame-level prediction for the same output. This conclusion can be generalized to multi-instance learning as well. However, bag prediction and instance prediction are used to indicate different levels of prediction.

In a nutshell, the experiment results in DCASE 2021 suggest that overall performance did not improve by memory-controlled self-attention. In DCASE 2021 task4, the ranking criterion is an aggregation of PSDS-scenario1 and PSDS-scenario2, and it can be obtained by two different systems. This criterion was decided by the compelling fact that different systems can be adopted in different settings by the same user. Our work suggests that specifying different pooling enables the SED system to lay emphasis on different scenarios, and linear softmax pooling could be a good choice when the evaluation criterion is not known prior. The effectiveness of our proposed adaptive controlled strategy has also been confirmed in a supervised model. Since the largest environment sound dataset Audio-Set was annotated with strong labels recently [47], our future work will focus on validating our model in a strongly labeled audio set.

6. Conclusions

In this study, we built a memory-controlled sequential self-attention mechanism on top of a CRNN model to develop a sound event detection system and investigated how well that attention mechanism could improve SED. DCASE 2021 datasets and DACSE 2017 datasets were used to test our proposed detection system. Evaluations were performed with various metrics. We found that memory-controlled self-attention can improve performance in PSDS scenario 2 and overall performance in real-life scenarios, say, in DCASE 2017. Our strategy for adaptively choosing an attention width was also successful: it forms a better bottleneck hidden state feature representation by taking appropriate length of context into consideration. The second major finding is that in cases of aggregating frame-level probability to form recording-level predictions, adding an attention layer on top of the CRNN is needless repetition: it fuses with the input of the following pooling layer, leading to a degradation in performance. While we draw this conclusion specifically for SED, it can be generalized to multi-instance learning in other domains. Moreover, we investigated various pooling methods in two scenarios. In scenario 1, linear softmax achieved the highest PSDS. Average, linear softmax, and auto pooling are more suitable for scenario 2. Linear softmax is by no means the best choice. It only performed nearly as well as the best methods in various evaluation metrics, but importantly, it does not require any parameters to be trained. This conclusion will be of interest to those enrolling in DCASE challenges.

Author Contributions: Conceptualization, M.W. and Y.Y.; software, X.S. and Y.Y.; writing—review and editing, M.W. and X.S.; formal analysis, M.W. and H.Q.; funding acquisition, M.W. and H.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of China (grants 62071135), the Natural Science Foundation of Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education (Guilin University of Electronic Technology): CRKL20011 and the Innovation Project of GUET Graduate Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found on http://dcase.community/ (accessed on 1 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mesaros, A.; Diment, A.; Elizalde, B.; Heittola, T.; Vincent, E.; Raj, B.; Virtanen, T. Sound event detection in the DCASE 2017 challenge. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 992–1006. [CrossRef]
2. Serizel, R.; Turpault, N.; Eghbal-Zadeh, H.; Shah, A.P. Large-scale weakly labeled semi-supervised sound event detection in domestic environments. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), Surrey, UK, 19–23 November 2018. Available online: https://hal.inria.fr/hal-01850270 (accessed on 1 November 2018).
3. Turpault, N.; Serizel, R.; Salamon, J.; Shah, A.P. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, New York, NY, USA, 25–26 October 2019. Available online: https://hal.inria.fr/hal-02160855 (accessed on 16 July 2019).
4. Foggia, P.; Petkov, N.; Saggese, A.; Strisciuglio, N.; Vento, M. Reliable detection of audio events in highly noisy environments. *Pattern Recognit. Lett.* **2015**, *65*, 22–28. [CrossRef]
5. Crocco, M.; Cristani, M.; Trucco, A.; Murino, V. Audio surveillance: A systematic review. *ACM Comput. Surv. (CSUR)* **2016**, *48*, 1–46. [CrossRef]
6. Koutini, K.; Eghbal-zadeh, H.; Dorfer, M.; Widmer, G. The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), La Coruña, Spain, 2–6 September 2019; pp. 1–5. [CrossRef]
7. Adavanne, S.; Politis, A.; Virtanen, T. Multichannel Sound Event Detection Using 3D Convolutional Neural Networks for Learning Inter-channel Features. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7. [CrossRef]
8. Virtanen, T.; Plumbley, M.D.; Ellis, D. *Computational Analysis of Sound Scenes and Events*; Springer: Berlin/Heidelberg, Germany, 2018; Chapter 2, pp. 13–25.
9. Cao, Y.; Kong, Q.; Iqbal, T.; An, F.; Wang, W.; Plumbley, M.D. Polyphonic sound event detection and localization using a two-stage strategy. *arXiv* **2019**, arXiv:1905.00268.
10. Adavanne, S.; Pertilä, P.; Virtanen, T. Sound Event Detection Using Spatial Features and Convolutional Recurrent Neural Network. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, New Orleans, LA, USA, 5–9 March 2017.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
12. Li, N.; Liu, S.; Liu, Y.; Zhao, S.; Liu, M.; Zhou, M. Close to human quality TTS with Transformer. *arXiv* **2018**, arXiv:1809.08895.
13. Dong, L.; Shuang, X.; Bo, X. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In Proceedings of the ICASSP 2018—2018 IEEE Interna-tional Conference on Acoustics, Speech and Signal Pro-cessing (ICASSP), Calgary, AB, Canada, 15–20 April 2018.
14. Wang, Y.; Metze, F. A first attempt at polyphonic sound event detection using connectionist temporal classification. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2986–2990.
15. Chen, Y.; Jin, H. Rare Sound Event Detection Using Deep Learning and Data Augmentation. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019; pp. 619–623. [CrossRef]
16. Kim, B.; Ghaffarzadegan, S. Self-supervised Attention Model for Weakly Labeled Audio Event Classification. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), La Coruna, Spain, 2–6 September 2019.
17. Kong, Q.; Yong, X.; Wang, W.; Plumbley, M.D. Audio Set Classification with Attention Model: A Probabilistic Perspective. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2018, Seoul, Korea, 15–20 April 2018.
18. Wang, J.; Li, S. Self-attention mechanism based system for DCASE2018 challenge Task1 and Task4. In Proceedings of the DCASE Challenge, Surrey, UK, 19-20 November 2018; pp. 1–5.
19. Sukhbaatar, S.; Grave, E.; Bojanowski, P.; Joulin, A. Adaptive Attention Span in Transformers. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
20. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.

21. Pankajakshan, A.; Bear, H.L.; Subramanian, V.; Benetos, E. Memory Controlled Sequential Self Attention for Sound Recognition. *arXiv* **2020**, arXiv:2005.06650.
22. Kim, B.; Pardo, B. Sound Event Detection Using Point-Labeled Data. In Proceedings of the 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 20–23 October 2019.
23. Kim, B.; Pardo, B. A human-in-the-loop system for sound event detection and annotation. *ACM Trans. Interact. Intell. Syst. Tiis* **2018**, *8*, 13. [CrossRef]
24. Kumar, A.; Raj, B. Audio event detection using weakly labeled data. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 1038–1047.
25. Frederic, F.; Annamaria, M.; Daniel, E.; Eduardo, F.; Magdalena, F.; Benjamin, E. In Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021), Online, 15–19 November 2021.
26. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv* **2017**, arXiv:1703.01780.
27. Bilen, Ç.; Ferroni, G.; Tuveri, F.; Azcarreta, J.; Krstulović, S. A Framework for the Robust Evaluation of Sound Event Detection. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 61–65. [CrossRef]
28. McFee, B.; Salamon, J.; Bello, J.P. Adaptive Pooling Operators for Weakly Labeled Sound Event Detection. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 2180–2193. [CrossRef]
29. Wang, Y.; Li, J.; Metze, F. A Comparison of Five Multiple Instance Learning Pooling Functions for Sound Event Detection with Weak Labeling. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2–17 May 2019; pp. 31–35. [CrossRef]
30. Lin, L.; Wang, X.; Liu, H.; Qian, Y. Specialized Decision Surface and Disentangled Feature for Weakly-Supervised Polyphonic Sound Event Detection. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1466–1478. [CrossRef]
31. Kong, Q.; Yu, C.; Xu, Y.; Iqbal, T.; Wang, W.; Plumbley, M.D. Weakly Labelled AudioSet Tagging With Attention Neural Networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 1791–1802. [CrossRef]
32. Dinkel, H.; Wu, M.; Yu, K. Towards Duration Robust Weakly Supervised Sound Event Detection. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 887–900. [CrossRef]
33. Su, T.-W.; Liu, J.-Y.; Yang, Y.-H. Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 791–795.
34. Kumar, A.; Khadkevich, M.; Fugen, C. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 326–330.
35. Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M.D. Large-scale weakly supervised audio classification using gated convolutional neural network. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 121–125.
36. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
37. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
38. Mesaros, A.; Heittola, T.; Virtanen, T. Metrics for polyphonic sound event detection. *Appl. Sci.* **2016**, *6*, 162. [CrossRef]
39. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
40. Nam, H.; Kim, S.H.; Park, Y.H. FilterAugment: An Acoustic Environmental Data Augmentation Method. *arXiv* **2021**, arXiv:2110.03282.
41. Adavanne, S.; Virtanen, T. A report on sound event detection with different binaural features. *arXiv* **2017**, arXiv:1710.02997.
42. Mesaros, A.; Heittola, T.; Diment, A.; Elizalde, B.; Shah, A.; Vincent, E.; Raj, B.; Virtanen, T. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), Munich, Germany, 16–17 November 2017; pp. 85–92.
43. Zheng, X.; Song, Y.; McLoughlin, I.; Liu, L.; Dai, L.-R. An Improved Mean Teacher Based Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 356–360. [CrossRef]
44. Wang, Y.W.; Chen, C.P.; Lu, C.L.; Chan, B.C. Semi-Supervised Sound Event Detection Using Multiscale Channel Attention and Multiple Consistency Training. In Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021), Online, 15–19 November 2021.
45. Dinkel, H.; Cai, X.; Yan, Z.; Wang, Y.; Zhang, J.; Wang, Y. A lightweight approach for semi-supervised sound event detection with unsupervised data augmentation. In Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021), Online, 15–19 November 2021.

46. Kong, Q.; Xu, Y.; Wang, W.; Plumbley, M.D. A joint detection-classification model for audio tagging of weakly labelled data. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 641–645. [CrossRef]
47. Hershey, S.; Ellis, D.P.; Fonseca, E.; Jansen, A.; Liu, C.; Moore, R.C.; Plakal, M. The Benefit of Temporally-Strong Labels in Audio Event Classification. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021.