Power Pooling: An Adaptive Pooling Function for Weakly Labelled Sound Event Detection

Yuzhuo Liu[1,2], Hangting Chen[1,2], Yun Wang[3], Pengyuan Zhang*[1,2]
[1]*Key Laboratory of Speech Acoustics and Content Understanding*
Institute of Acoustics, Chinese Academy of Sciences, China
[2]*University of Chinese Academy of Sciences, Beijing, China*
[3]*Facebook AI, Menlo Park, CA, USA*
{liuyuzhuo,chenhangting,zhangpengyuan}@hccl.ioa.ac.cn, yunwang@fb.com

***Abstract*—Access to large corpora with strongly labelled sound events is expensive and difficult in engineering applications. Many researches turn to address the problem of how to detect both the types and the timestamps of sound events with weak labels that only specify the types. This task can be treated as a multiple instance learning (MIL) problem, and a key to it in the sound event detection (SED) task is the design of a pooling function. The linear softmax pooling function achieves state-of-the-art performance since it can vary both the signs and the magnitudes of gradients. However, linear softmax pooling cannot flexibly deal with sound events of different time scales. In this paper, we propose a power pooling function which can automatically adapt to various sound events. By adding a trainable parameter to each event, power pooling can provide more accurate gradients for frames in a clip than other pooling functions. On both weakly supervised and semi-supervised SED datasets, the proposed power pooling function outperforms linear softmax pooling on both coarse-grained and fine-grained metrics. Specifically, it improves the event-based $\mathbf{F_1}$ score by 11.4% and 10.2% relatively on the two datasets. While this paper focuses on SED applications, the proposed method can be applied to MIL tasks in other domains.**

I. Introduction

Sound event detection (SED) aims to identify the categories and timestamps of target sound events in continuous audio recordings. Some studies only focus on the categories of sound events present (*audio tagging*), while this paper pays more attention to the detection of onset and offset time (*localization*). Traditional SED models are often trained from data with strong labels, which contain the categories and timestamps of each sound event occurrence [1]–[4]. However, in real-world applications, such as noise monitoring [5], surveillance systems [6], machine condition monitoring [7], and multimedia indexing [8], acquiring such strong labels can incur a high cost. Google has released a large-scale weakly labelled dataset (AudioSet) [9] with clip-level annotations. AudioSet has led researchers to pay more attention to weakly labelled SED, *i.e.* when fine-grained timestamps are unavailable.

Weakly supervised tasks can be addressed as a multiple instance learning (MIL) [10] problem. As strong labels are expensive, the studies of MIL are widely conducted in many domains, such as object detection [11]–[13]. In MIL, instances are grouped into bags. Only the bag-level annotations are available, but the label of each instance is unknown. The label convention between a bag and the instances in it obey the *standard multiple instance* (SMI) *assumption*: A bag annotation is positive if at least one instance is positive. As for SED, for each event type, an audio clip and its frames can be regarded as a bag and instances in the bag. A positive bag is a clip that contains the given type of event, which means that, it consists of at least one positive frame and may also contain negative frames; A negative bag, on the other hand, only consists of negative frames.

To improve the localization accuracy, considerable researches have made efforts to select positive instances more precisely. Many studies are devoted to the design of *pooling functions*: as shown in Fig. 1, the pooling function calculates the clip-level probability of an event type as a weighted average of the frame-level probabilities, and also serves to back-

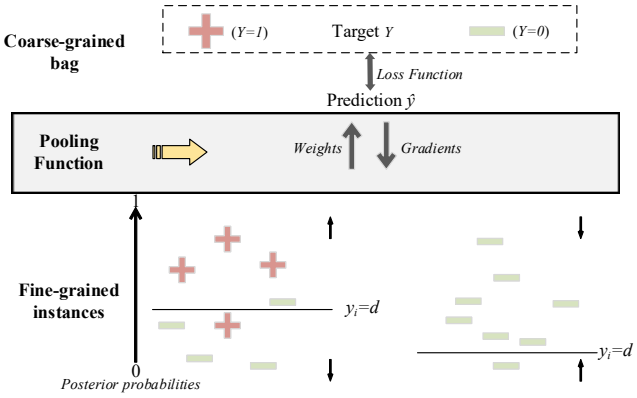

Fig. 1. Pooling function in a MIL system for SED with weak labels. A pooling function produces weights for fine-grained (frame-level) probabilities y_i to obtain a coarse-grained (clip-level) posterior probability $\hat{y}$, and generates fine-grained gradients from a coarse-grained loss. The plus signs (red) indicate instances (frames) whose ground truth is positive, and the minus signs (green) indicate instances whose ground truth is negative. d is the ideal threshold at which the gradients change signs. Arrows indicate the directions of the gradients.

This work is supported by the National Natural Science Foundation of China (No. 62071461).

*Pengyuan Zhang is the corresponding author.

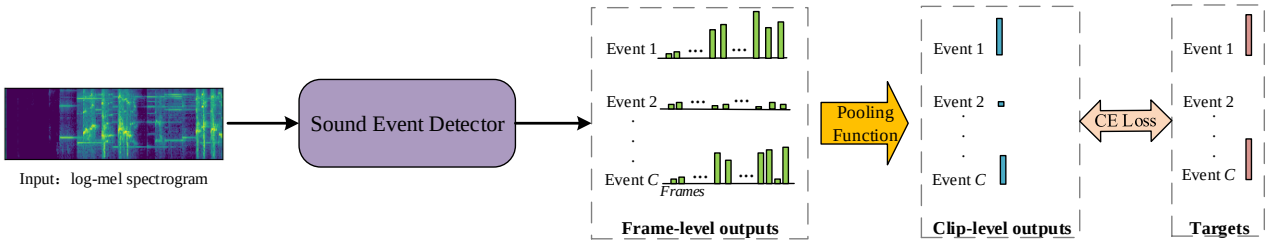

Fig. 2. Overview of a weakly supervised SED system. C is the number of target classes. "CE loss" is the cross-entropy loss function.

propagate gradients from the clip-level loss function to the frames. Ideally, the pooling function should produce weights and gradients discriminative enough. *"Max" pooling* [14] assigns zero weights to non-maximizing frames, which produces zero gradients and leads to difficult optimization. *Average pooling* [15] weights all frames equally and produces positive gradients for all frames, which leads to undesired gradients for negative instances in the positive bags. *Linear softmax pooling* [16], *exponential softmax pooling* [17], and *attention pooling* [18] assign different weights to frames with varying the magnitudes of the their gradients. Wang *et al.* [19] compared the above five pooling functions, and demonstrated that linear softmax pooling was the best at localizing sound events because it could produce positive gradients for some frames and negative gradients for others. McFee *et al.* [20] developed a family of adaptive pooling operators named *auto-pool* which could achieve a similar effect. He *et al.* [21] proposed a hierarchical pooling structure, which has shown remarkable performance combined with linear, exponential softmax, and attention pooling.

It remains challenging, however, to predict the onsets and offsets of sound events using these pooling functions. A core reason is that environmental sounds in general have less structure in comparison to speech and music. Diverse independent sources (*e.g.* animals, vehicles, electrical appliances) produce acoustic events with considerable variability. A key factor that causes variability is the varied durations of different events, and these pooling functions cannot adapt to such variability.

To mitigate the above issues, we design a simple but effective pooling function termed as *power pooling*. We use a power function of the predicted frame-level probability as the weight for each frame, and set the exponent as a trainable parameter to automatically generate an optimal threshold which is used to partition positive and negative gradients. The power pooling provides variable gradient directions for frame-level predictions with a flexible threshold. The trainable power parameter can also be made class-wise dependent, thus adapting to various sound events with different acoustic characteristics. We evaluate the proposed method on DCASE 2017 Task 4 [22] and DCASE 2019 Task 4 [23] datasets. The former is a weakly labelled SED dataset with longer events, while the latter is a semi-supervised SED dataset with shorter events. Our empirical results show that power pooling outperforms other pooling functions on all metrics.

II. Method

In this section, we first formulate the problem of weakly supervised SED. Then, we introduce the definition and limitations of the linear softmax pooling. Next, we develop a power pooling function which can adaptively generate proper weights and gradients for various sound events. Finally, we explain the similarities and differences between power pooling and auto pooling.

A. Formulation of weakly supervised SED

Polyphonic SED with C event categories can be regarded as C binary MIL problems. As such, we will only consider one event category hereafter.

Each training clip can be regarded as a bag (X, Y), where $X = [x_1, \ldots, x_m]$ is a sequence of instances (frames), and what we have is the label of the bag $Y \in \{0, 1\}$. The purpose is to learn a SED system which can output fine-grained (frame-level) and coarse-grained (clip-level) predictions of event probabilities simultaneously from such weakly labelled data.

Fig. 2 gives an overview of the weakly supervised SED system in this work. The system accepts the log-scaled Mel spectrograms $X^{m \cdot k}$ where m and k are numbers of frames and Mel filters, uses the detector to generate a frame-level probability $y_i \in [0, 1]$ for each instance x_i, uses the pooling function to aggregate y_i into a clip-level prediction $\hat{y} \in [0, 1]$. The pooling function assigns a weight w_i to each frame and taking the weighted average:

$$\hat{y} = \frac{\sum_i y_i \cdot w_i}{\sum_i w_i}. \tag{1}$$

To train the SED system, the cross-entropy (CE) loss function is minimized between the clip-level prediction $\hat{y}$ and the label Y. During back-propagation, the pooling function determines the gradient received by each instance, and the gradients should have appropriate signs. To evaluate the system, we apply a threshold γ on both y_i and $\hat{y}$ to generate binary predictions.

B. Linear softmax pooling

A pooling function should generally assign larger weights to frames with larger predicted probabilities, which is in order to

TABLE I
THE DIRECTIONS OF CLIP-LEVEL AND FRAME-LEVEL GRADIENTS IN LINEAR SOFTMAX POOLING ($\theta = 1/2$) AND POWER POOLING ($\theta = n/(n+1)$).

Target	Clip-level	Condition	Frame-level
positive ($Y = 1$)	$\hat{y} \to 1$	$y_i > \theta \cdot \hat{y}$	$y_i \to 1$
		$y_i < \theta \cdot \hat{y}$	$y_i \to 0$
negative ($Y = 0$)	$\hat{y} \to 0$	$y_i > \theta \cdot \hat{y}$	$y_i \to \theta \cdot \hat{y}$
		$y_i < \theta \cdot \hat{y}$	$y_i \to \theta \cdot \hat{y}$

conform to the SMI assumption. The state-of-the-art (SOTA) linear softmax pooling uses a weight w_i equal to y_i:

$$\hat{y} = \frac{\sum_i (y_i \cdot y_i)}{\sum_i y_i}, \tag{2}$$

and its gradient is

$$\frac{\partial \hat{y}}{\partial y_i} = \frac{2y_i - \hat{y}}{\sum_j y_j}. \tag{3}$$

This gradient is positive if and only if $y_i > \hat{y}/2$. For positive clips ($Y = 1$), this causes "larger" frame-level probabilities to increase and "smaller" frame-level probabilities to decrease, thereby yielding clear boundaries of event occurrences. The threshold between "larger" and "smaller" probabilities is given by $d = \hat{y}/2$. For negative clips ($Y = 0$), the gradients pushes all frame-level probabilities toward $\hat{y}/2$. Considering that this threshold is smaller than $\hat{y}$, all the frame-level probabilities will converge to 0 as desired after enough iterations. The movements of the frame-level probabilities are listed in Table I as well as depicted in Fig. 1.

Define $\theta = d/\hat{y}$ as the ratio between the threshold at which the gradient changes signs and the clip-level predicted probability. In linear softmax pooling, θ is fixed at $1/2$. In reality, however, it may be desirable to have a different θ for different event categories. For example, we may want to boost the predicted probabilities of more frames when a clip contains a type of event that usually lasts a long time (*e.g.* vacuum cleaner), and boost fewer frames when the type of event is usually transient (*e.g.* dog bark). This motivated us to propose *power pooling*.

C. Power pooling

Without changing the pattern of how predictions move in Table I, we hope to make the ratio θ variable by adding a small number of trainable parameters on the basis of the linear softmax pooling function. We propose to determine the pooling weights of the frame-level probability using the power function $f(x) = x^n$, where n is the single trainable parameter. In other words, we use the n-th power of the frame-level probabilities $(y_i)^n$ as the weights for pooling:

$$\hat{y} = \frac{\sum_i y_i \cdot (y_i)^n}{\sum_i (y_i)^n}. \tag{4}$$

We call this pooling function *power pooling*. To conform to the SMI assumption, $w_i = (y_i)^n$ must be an increasing function in y_i, therefore the power parameter n must be positive.

The gradient of the power pooling function is:

$$\frac{\partial \hat{y}}{\partial y_i} = \frac{(n+1) \cdot (y_i)^n - n \cdot (y_i)^{n-1} \cdot \hat{y}}{\sum_j (y_j)^n}. \tag{5}$$

During back-propagation, frame-level probabilities will move in the same pattern as in Table I, while the threshold is given by $d = \theta \cdot \hat{y}$ with $\theta = n/(n+1) \in (0, 1)$. The gradient still exhibits different signs for different frames.

Power pooling inherits the advantage of linear softmax pooling at localizing sound events. In addition, the learnable power parameter n allows it to approach either max pooling (as $n \to +\infty$) or average pooling (as $n \to 0$). When n is fixed to 1, power pooling reduces to linear softmax pooling.

It is desirable to make the power n depend on the event category: for long-lasting events we prefer to choose a smaller n, which results in a lower threshold and boosts more frames; for transient events we would do the opposite. When the power n gets too large, however, power pooling can suffer from the same problem of zero gradients as max pooling. To avoid this problem, we add a regularization term of $\lambda \sum_c n_c^2$ to the loss function, where n_c is the power parameter for the event category c. Thus, the formula of final loss function is

$$\begin{aligned} L &= CE(Y, \hat{y}) + \lambda \sum_c n_c^2 \\ &= CE\left(Y, \frac{\sum_i y_i \cdot (y_i)^n}{\sum_i (y_i)^n}\right) + \lambda \sum_c n_c^2. \end{aligned} \tag{6}$$

D. Relationship with auto-pool

The term "auto-pool" encompasses three pooling operators: auto pooling (*Auto*), constrained auto pooling (*CAP*) and regularized auto pooling (*RAP*). Auto pooling adds a trainable parameter α to the exponential softmax pooling function; its formula and gradient are:

$$\hat{y} = \frac{\sum_i y_i \cdot \exp(\alpha \cdot y_i)}{\sum_i \exp(\alpha \cdot y_i)}, \tag{7}$$

$$\frac{\partial \hat{y}}{\partial y_i} = (1 - \alpha\hat{y} + \alpha y_i) \cdot \frac{\exp(\alpha y_i)}{\sum_j \exp(\alpha y_j)}. \tag{8}$$

Auto pooling allows exponential softmax to approach either max pooling (as $\alpha \to +\infty$) or average pooling (as $\alpha \to 0$). To avoid approaching the max pooling behavior too much, *CAP* and *RAP* are proposed to restrict auto pooling. *CAP* imposes an upper limit of $\ln(m-1)$ on α, where m is the number of frames in a clip, so that no frame may receive more than half of the total weight. *RAP* adds a quadratic regularization term $\lambda\alpha^2$ to the objective function similar to power pooling.

We would like to point out that the gradient in auto pooling is always positive when $\alpha \in (-1, 1)$, therefore it is less discriminative between positive and negative instances in a positive bag than power pooling. Moreover, in auto pooling, frames with near-zero probabilities still get weights close to 1 due to the $\exp$ operator, and may unwantedly dominate the weighted average. This is unlike power pooling, where frames with near-zero probabilities get near-zero weights.

TABLE II
THE WINDOW LENGTHS OF MEDIAN FILTERS FOR THE EVENT CATEGORIES OF DCASE 2019 TASK 4.

Event Category	Length/s	Event Category	Length/s
Alarm/bell/ringing	0.21	Electric shaver/toothbrush	2.24
Blender	1.20	Frying	2.82
Cat	0.48	Running water	1.98
Dishes	0.20	Speech	0.49
Dog	0.26	Vacuum cleaner	2.92

III. EXPERIMENTAL SETUP

A. Dataset

We carry out experiments to compare the performance of power pooling with other pooling functions on DCASE 2017 [22] and DCASE 2019 [23] datasets. Both datasets contain realistic clips taken from AudioSet [9], and DCASE 2019 has an additional subset of synthetic recordings. Most clips have a duration of 10 seconds (a few clips are shorter), and multiple audio events may occur at the same time.

DCASE 2017 [22]: The dataset of Task 4 of the DCASE 2017 challenge is composed of 17 types of "warning" and "vehicle" sounds. We take the weakly labelled training set (51,172 clips) and the strongly labelled public test set (488 clips). The mean duration of each type of event varies from 4 s to 10 s, covering more than 40% of the clips.

DCASE 2019 [23]: The DCASE 2019 dataset focuses on 10 types of sound events in domestic environments. It consists of three training sets (synthetic strongly labelled: 2,045 clips, weakly labelled: 1,578 clips, unlabelled: 14,412 clips) and a validation set (1,168 clips). The mean duration of each type of event varies from 0.5 s to 5 s, covering less than 50% of the clips.

The two datasets involve relatively long and short events, respectively. They also allow us to verify the performance of the pooling functions in a pure weakly supervised scenario as well as a semi-supervised scenario.

B. Evaluation

The performance of systems is measured with fine-grained and coarse-grained F_1 score:

$$F_1 = \frac{P + R}{2 \cdot P \cdot R}, \tag{9}$$

$$P = \frac{TP}{TP + FP}, \tag{10}$$

$$R = \frac{TP}{TP + FN}, \tag{11}$$

where TP, FP, and FN are the number of true positives, false positives and false negatives. The F_1 score balances precision (P) and recall (R). For fine-grained evaluation, we compute both event-level and segment-level metrics. The former treats each event occurrence as an instance with a 200 ms collar on onsets and a collar of 200 ms or 20% of the event length on offsets; the latter treats each 1 s segment as an instance. For coarse-grained evaluation, we compute clip-level metrics treating each entire 10 s clip as an instance. We use the macro-average to aggregate the metrics across event categories. The evaluation details can be found in [24].

TABLE III
THE DEFINITIONS AND GRADIENTS OF POOLING FUNCTIONS. m IS THE NUMBER OF FRAMES IN A CLIP. α AND n ARE TRAINABLE PARAMETERS.

Pooling Function	Definition	Gradient
Max	$\hat{y} = \max_i y_i$	$\frac{\partial \hat{y}}{\partial y_i} = \begin{cases} 1, & \text{if } y_i = \hat{y} \\ 0, & \text{otherwise} \end{cases}$
Average	$\hat{y} = \frac{1}{m} \sum_i y_i$	$\frac{\partial \hat{y}}{\partial y_i} = \frac{1}{m}$
Exp	$\hat{y} = \frac{\sum_i y_i \cdot \exp(y_i)}{\sum_i \exp(y_i)}$	$\frac{\partial \hat{y}}{\partial y_i} = (1 - \hat{y} + y_i) \cdot \frac{\exp(y_i)}{\sum_j \exp(y_j)}$
Attention	$\hat{y} = \frac{\sum_i y_i \cdot w_i}{\sum_i w_i}$	$\frac{\partial \hat{y}}{\partial y_i} = \frac{w_i}{\sum_j w_j}$
Auto	$\hat{y} = \frac{\sum_i y_i \cdot \exp(\alpha y_i)}{\sum_i \exp(\alpha y_i)}$	$\frac{\partial \hat{y}}{\partial y_i} = (1 - \alpha \hat{y} + \alpha y_i) \cdot \frac{\exp(\alpha y_i)}{\sum_j \exp(\alpha y_j)}$
Linear	$\hat{y} = \frac{\sum_i y_i \cdot y_i}{\sum_i y_i}$	$\frac{\partial \hat{y}}{\partial y_i} = \frac{2y_i - \hat{y}}{\sum_j y_j}$
Power	$\hat{y} = \frac{\sum_i y_i \cdot (y_i)^n}{\sum_i (y_i)^n}$	$\frac{\partial \hat{y}}{\partial y_i} = \frac{(n+1) \cdot (y_i)^n - n \cdot (y_i)^{n-1} \cdot \hat{y}}{\sum_j (y_j)^n}$

C. Model architecture and post-processing

On DCASE 2017, the data pre-processing and model architecture are nearly the same as in [19]. In a nutshell, the input Mel features have 400 frames and 64 frequency bins, and the model structure contains 3 convolutional blocks, 2 bidirectional gated recurrent unit (BiGRU) layers, and 1 dense layer. We add a batch norm layer to each convolutional block.

On DCASE 2019, we apply a semi-supervised framework since the dataset contains unlabelled data. The data pre-processing and backbone model are based on [25]. We adopt the popular *mean-teacher* [26] architecture and a feature extractor implemented as a convolutional recurrent neural network. Notably, as DCASE 2019 contains synthetic strong labelled data and unlabelled data, a frame-level CE loss and a consistency loss are applied during training [25]. Furthermore, the following optimizations are performed: First, we augment the data by shifting input features along the time axis, sampling the shift from a normal distribution with zero mean and a standard deviation of 16 frames. Second, we adopt the hyperparameters of the feature extractor in [27]. Third, we apply a set of median filters on the frame-level predicted probabilities, using window lengths proportional to the average duration of each event category. The specific window lengths can be found in Table II.

The regularization hyperparameter λ for the power parameters is set to 10^{-4} for DCASE 2017 and 0 for DCASE 2019. The threshold γ is tuned to optimize the clip-level F_1 score on DCASE 2017, and is fixed to 0.5 on DCASE 2019, which follows the strategy in [19], [27].

D. Baseline pooling functions

We take the following pooling functions as baselines to compare against the proposed power pooling:

TABLE IV
DETAILED RESULTS ON DCASE 2017 TASK 4 AND DCASE 2019 TASK 4. WE INDICATE IN BOLD THE BEST F_1 SCORES ACROSS ALL SYSTEMS, AS WELL AS THE BEST APART FROM POWER POOLING.

Pooling Function	**DCASE 2017**							**DCASE 2019**						
	Event-level			Segment-level			Clip-level	Event-level			Segment-level			Clip-level
	F_1	Precision	Recall	F_1	Precision	Recall	F_1	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Max [14]	0.094	0.169	0.073	0.372	0.567	0.300	0.465	0.256	0.381	0.201	0.488	0.836	0.362	0.609
Average [15]	0.165	0.147	0.196	0.450	0.425	0.515	0.516	0.171	0.158	0.201	0.564	0.499	0.675	0.597
Exp [17]	0.166	0.154	0.187	0.466	0.472	0.481	0.521	0.187	0.190	0.203	0.569	0.559	0.654	0.611
Attention [18]	0.130	0.139	0.171	0.434	0.481	0.470	0.527	0.320	0.359	0.300	**0.600**	0.688	0.547	0.386
Auto [20]	0.169	0.144	0.217	0.457	0.425	0.535	0.536	0.218	0.288	0.180	0.597	0.755	0.526	**0.655**
CAP [20]	0.164	0.152	0.199	0.468	0.447	0.521	**0.544**	0.188	0.217	0.171	0.598	0.628	0.601	0.641
RAP 10^{-2} [20]	**0.176**	0.147	0.233	0.464	0.411	0.561	0.532	0.177	0.189	0.179	0.584	0.544	0.668	0.639
RAP 10^{-3} [20]	0.165	0.145	0.202	0.464	0.432	0.529	0.534	0.172	0.175	0.181	0.586	0.530	0.682	0.640
RAP 10^{-4} [20]	0.158	0.132	0.206	0.455	0.410	0.539	0.526	0.178	0.229	0.151	0.537	0.602	0.533	0.516
Linear [19]	0.162	0.178	0.161	**0.471**	0.542	0.451	0.535	**0.343**	0.431	0.292	0.583	0.738	0.498	**0.655**
Power	**0.196**	0.168	0.248	**0.480**	0.460	0.537	**0.545**	**0.378**	0.437	0.340	**0.624**	0.752	0.547	**0.694**

TABLE V
COMPARISON OF THE F_1 SCORES OF OUR POWER POOLING WITH THREE POPULAR POOLING FUNCTIONS ACROSS THE 10 EVENT CATEGORIES OF DCASE 2019 TASK 4. BEST RESULTS ARE SHOWN IN BOLD.

Category	**Event-level** F_1				**Segment-level** F_1				**Clip-level** F_1			
	Attention	Auto	Linear	Power	Attention	Auto	Linear	Power	Attention	Auto	Linear	Power
Alarm/bell/ringing	0.403	0.264	0.460	**0.473**	0.735	0.735	0.798	**0.822**	0.474	0.755	0.816	**0.840**
Blender	0.325	0.321	0.319	**0.462**	**0.560**	0.530	0.516	0.552	0.336	0.571	0.549	**0.614**
Cat	0.364	0.070	**0.442**	0.407	0.519	**0.594**	0.525	0.567	0.082	**0.738**	0.688	0.734
Dishes	**0.204**	0.107	0.167	0.199	0.429	**0.460**	0.376	0.456	0.041	0.482	0.524	**0.581**
Dog	**0.230**	0.060	0.128	**0.230**	0.621	0.628	0.638	**0.648**	0.441	0.699	**0.716**	**0.716**
Electric shaver/toothbrush	0.352	0.447	**0.396**	0.388	**0.669**	0.608	0.551	0.598	0.636	0.646	0.617	**0.653**
Frying	**0.316**	0.260	0.291	0.301	**0.543**	0.489	0.411	0.514	0.482	0.527	0.465	**0.556**
Running water	0.174	**0.269**	0.189	0.212	0.451	**0.584**	0.482	0.519	0.271	**0.631**	0.580	0.613
Speech	0.421	0.101	0.471	**0.489**	**0.828**	0.814	0.807	0.825	0.482	**0.895**	0.880	0.894
Vacuum cleaner	0.416	0.284	0.565	**0.623**	0.639	0.532	0.731	**0.736**	0.604	0.603	0.710	**0.743**

- Three classic pooling functions: *max*, *average*, and *exponential softmax* (*Exp* for short);
- The popular *attention* pooling;
- A family of adaptive pooling functions: *Auto*, *CAP*, and *RAP*;
- The state-of-the-art *linear softmax* pooling.

Table III lists the definitions and the gradients of all the pooling functions.

IV. RESULTS AND DISCUSSION

In this section, we first describe the marco-average results on two datasets and the detailed results of each sound events in DCASE 2019 dataset. We then visualize how power pooling improve the localization performance in an audio clip. Finally, we show the value of parameter n_c for sound events in two datasets to illustrate that power pooling adaptively generates proper n_c for sound events with various time scales.

A. Results on test datasets

Table IV compares power pooling with the baseline pooling functions on the DCASE 2017 and DCASE 2019 datasets. The impact of pooling functions appears to be similar on the weakly labelled dataset and the semi-supervised dataset. Power pooling achieves the highest F_1 at all the three levels on both datasets. For event-level evaluation, which is most relevant for sound event localization, power pooling achieves F_1 scores of 0.196 and 0.378 on the two datasets. Compared with the best scores of the baseline pooling functions (0.176, 0.343), power pooling shows an absolute improvement of 2% and 3.5%, and a relative improvement of 11.4% and 10.2%. As for the segment-level and clip-level metrics, linear softmax pooling, attention pooling and two of the auto pooling functions achieve the highest F_1 scores among the baseline functions (0.471, 0.600, 0.544, 0.655) on the two datasets, while models with power pooling outperform the above results with an absolute improvement from 0.1% to 4%. The experimental result demonstrates that the power pooling benefits both audio tagging and localization, indicating that it yields proper weights and gradients.

Comparing with the baseline linear softmax pooling, we can attribute the superior performance of power pooling to its ability to significantly increase the recall score while maintaining comparable precision. Power pooling overcomes the defect of linear softmax pooling that it tends to produce false negatives [19].

Table V shows the F_1 scores of the 10 event categories in DCASE 2019. Here, we compare our power pooling function with the three pooling functions that generated the highest F_1 at a certain level in Table IV. On 8 out of the 10 event categories, power pooling achieves a better event-level F_1 score than linear softmax pooling, demonstrating that power pooling can find proper power parameters and achieve more accurate localization. Moreover, compared with the other three pooling functions, power pooling exhibits competitive performance

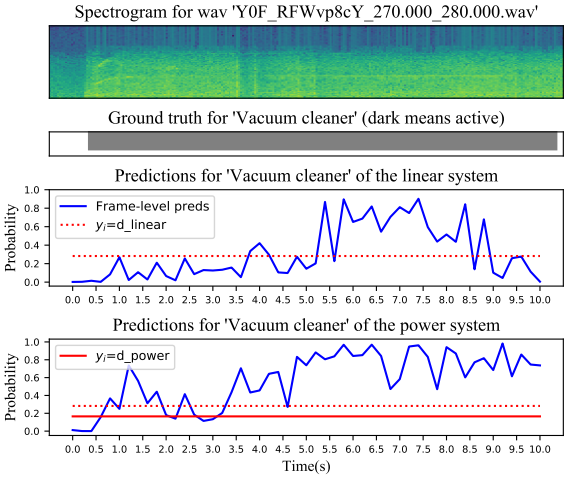

Fig. 3. Frame-level predictions of linear softmax and power pooling at epoch 40.

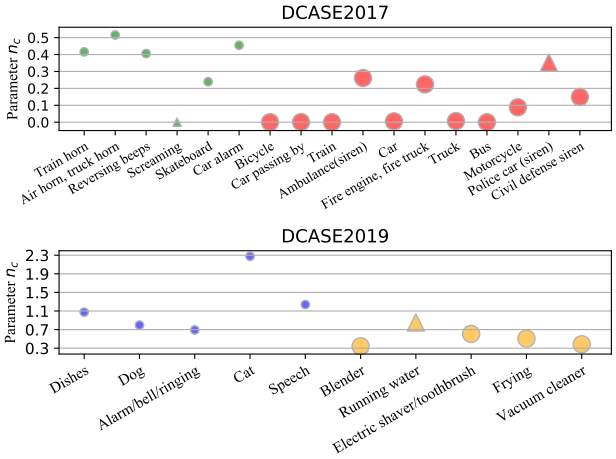

Fig. 4. The power parameter n_c for each event category in the final model.

(first or second highest) at all levels for all the ten events (except segment-level F_1 for "electric shaver/toothbrush"). If the practical purpose is to simultaneously detect multiple types of sound events, especially when their durations vary, power pooling appears to be the method of choice.

Table V also indicates that baseline functions have their own strengths, despite their difficult balance between evaluation granularities and poor flexibility toward various sound events. Attention pooling can benefit fine-grained detection in semi-supervised learning, as it achieves the highest event-level and segment-level F_1 scores for three and four event types, respectively. Nevertheless, it yields poor clip-level performance. The reason might be that an extra frame-level cross-entropy in semi-supervised learning leads attention pooling to pay more attention to fine-grained evaluation. Auto pooling can be useful on coarse-grained classification, as it obtains the highest clip-level F_1 scores on three event types of DCASE 2019, and high macro-average F_1 on both datasets at clip-level.

B. Visualization of localization improvement

Fig. 3 illustrates the predictions for the "vacuum cleaner" event on a weakly labelled training clip, produced by a linear softmax pooling system and a power pooling system after 40 epochs of training (we trained for 200 epochs in total). We also plot the actual temporal interval spanned by the event. For the power pooling system, we show the threshold between positive and negative gradients arising from both the power pooling function ($d_{\text{power}} = n/(n+1)\cdot\hat{y}$) and the linear softmax pooling function ($d_{\text{linear}} = 1/2\cdot\hat{y}$). The power parameter for the "vacuum cleaner" event is $n = 0.337 < 1$, yielding a lower threshold and allowing more frames to receive positive gradients. Compared with the linear softmax pooling system, more frames in the power pooling system receive a gradient in the correct direction, notably from 0.7 s to 3 s and 9 s to 10 s. This indicates how a more appropriate threshold between positive and negative gradients can help to correctly pinpoint the onset and offset of events.

C. Proper parameters n_c produced for various sound events

Fig. 4 shows the power parameters n_c of each event category in the final model. We sort the event categories horizontally by their average duration, and divide them roughly into shorter events (smaller symbols) and longer events (larger symbols). The durations of shorter and longer events in the DCASE 2017 dataset fall within 4–5 s and 6–10 s; for DCASE 2019, these durations fall within 0.5–1.1 s and 2–5 s. Except for a small portion of categories (marked by triangles), longer events tend to have smaller power parameters, making the power pooling approach average pooling; shorter events tend to have larger power parameters, making the power pooling approach max pooling. We even observe the power parameters n_c adapt to event durations across datasets: events in DCASE 2017 generally last longer than those in DCASE 2019; as a result, the parameters n_c are usually smaller than 0.5 for events in DCASE 2017, but are much larger in DCASE 2019. The experimental results agree well with the motivation.

V. Conclusion

This paper has proposed a practical power pooling function for weakly labelled SED. Power pooling overcomes the shortcoming of the SOTA linear softmax pooling, whose frame weights are determined by a fixed formula from its predicted probabilities. By adding only one learnable power parameter for each category, power pooling can automatically learn an appropriate threshold between positive and negative gradients for each event category. The learnable parameter allows power pooling to adapt to various sound events with different time scales. Our experimental results indicates that power pooling is much more preferred for weakly supervised SED, especially for polyphonic SED. Moreover, the power pooling function is generic enough to be applied to MIL problems in other domains.

REFERENCES

[1] E. Cakir, T. Heittola, H. Huttunen, *et al.*, "Polyphonic sound event detection using multi label deep neural networks", *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, July 2015, pp. 1–7.

[2] E. Cakir, E. C. Ozan, T. Virtanen, *et al.*, "Filterbank learning for deep neural network based polyphonic sound event detection", *IJCNN*, Vancouver, July 2016, pp. 3399–3406.

[3] G. Parascandolo, H. Huttunen, T. Virtanen, *et al.*, "Recurrent neural networks for polyphonic sound event detection in real life recordings", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, March 2016, pp. 6440–6444.

[4] E. Cakir and T. Virtanen, "End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input", *IJCNN*, Rio de Janeiro, Brazil, July 2018, pp. 2412–2418.

[5] J. P. Bello, C. Silva, O. Nov, *et al.*, "SONYC: A system for monitoring, analysis and mitigation of urban noise pollution", *Communications of the ACM*, 2018, pp. 68-77.

[6] Marco Crocco, Marco Cristani, Andrea Trucco, and Vittorio Murino. "Audio Surveillance: A Systematic Review", *ACM Comput. Surv.*, 48, 4, Article 52, May 2016, 46 pages. DOI:https://doi.org/10.1145/2871183.

[7] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, *et al.*, "Description and discussion on DCASE2020 challenge task2: unsupervised anomalous sound detection for machine condition monitoring", *arXiv e-prints: 2006.05822, 1â€“4*, June 2020, URL: https://arxiv.org/abs/2006.05822.

[8] S. Hershey, S. Chaudhuri, D. P. W. Eills, *et al.*, "CNN architectures for large-scale audio classification", *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, New Orleans, 2017: 131-135.

[9] J. F. Gemmeke, *et al.*, "Audio Set: An ontology and human-labelled dataset for audio events", *ICASSP*, New Orleans, LA, 2017, pp. 776–780, doi: 10.1109/ICASSP.2017.7952261.

[10] Jaume Amores, "Multiple instance classification: Review, taxonomy and comparative study", *Artificial Intelligence*, June 2013, vol 201, pp. 81-105.

[11] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-MIL: Continuation multiple instance learning for weakly supervised object detection", *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2199-2208.

[12] K. Yang, D. Li, and Y. Dou, "Towards precise End-to-End weakly supervised object detection network", *IEEE/CVF Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8372-8381.

[13] G. Cheng, J. Yang, D. Gao, L. Guo and J. Han, "High-Quality Proposals for Weakly Supervised Object Detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 5794-5804, 2020, doi: 10.1109/TIP.2020.2987161.

[14] T. Su, J. Liu and Y. Yang, "Weakly-supervised audio event detection using event-specific Gaussian filters and fully convolutional networks," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 791-795, doi: 10.1109/ICASSP.2017.7952264.

[15] A. Shah, A. Kumar, A. Hauptmann, and B. Raj, "A closer look at weak label learning for audio events", 2018, 10.13140/RG.2.2.29936.76806.

[16] A. Dang, T. H. Vu, and J.-C. Wang, "Deep learning for DCASE2017 challenge", Technical Report, *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Challenge*, 2017.

[17] J. Salamon, B. McFee, and P. Li, "DCASE 2017 submission: Multiple instance learning for sound event detection", Technical Report, *DCASE 2017 Challenge*, 2017.

[18] Q. Kong, *et al.*, "Audio Set classification with attention model: A probabilistic perspective", *ICASSP*, April 2018.

[19] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labelling", *ICASSP*, 2019, pp. 31–35.

[20] B. Mcfee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labelled sound event detection", *Audio, Speech, and Language Processing, IEEE/ACM Transactions*, 2018, pp. 2180–2193.

[21] K. He, Y. Shen, and W. Zhang, "Hierarchical pooling structure for weakly labelled sound event detection", *Proceedings of Interspeech)*, 2019, pp. 3624–3628.

[22] A. Mesaros, T. Heittola, A. Diment, *et al.*, "DCASE 2017 challenge setup: Tasks, datasets and baseline system", *DCASE 2017 Workshop*, 2017, pp. 85–92.

[23] R. Serizel, N. Turpault, E. Hamid, *et al.*, "Large-scale weakly labelled semi-supervised sound event detection in domestic environments", *DCASE 2018 workshop*, 2018, pp. 19–23.

[24] A. Mesaros, T. Heittola, T. Virtanen, *et al.*, "Metrics for polyphonic sound event detection", *Applied Sciences*, 2016, pp. 162–167.

[25] J. K. Lu, "Mean teacher convolution system for DCASE 2018 Task 4", Technical Report, *DCASE 2018 Challenge*, June 2018.

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

[27] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 Task 4", Technical Report, *DCASE 2019 Challenge*, June 2019.