arXiv:1905.10091v5 [cs.SD] 19 Sep 2019

Specialized Decision Surface and Disentangled Feature for Weakly-Supervised Polyphonic Sound Event Detection

Liwei Lin[1,2], Xiangdong Wang[1], Hong Liu[1], and Yueliang Qian[1]

***Abstract*—Sound event detection (SED) consists in recognizing the presence of sound events in the segment of audio and detecting their onset as well as offset. In this paper, we focus on two common problems on SED: how to carry out efficient weakly-supervised learning and how to learn better from the unbalanced dataset in which multiple sound events often occur in co-occurrence.**

We approach SED as a multiple instance learning (MIL) problem and utilize a neural network framework with different pooling modules to solve it. General MIL approaches includes two approaches: the instance-level approach and the embedding-level approach. Since the embedding-level approach tends to perform better than the instance-level approach in terms of bag-level classification but can not provide instance-level probabilities, we present how to generate instance-level probabilities for it. Moreover, we further propose a specialized decision surface (SDS) for the embedding-level attention pooling. We analyze and explained why an embedding-level attention module with SDS is better than other typical pooling modules from the perspective of the high-level feature space. As for the problem of unbalanced dataset and the co-occurrence of multiple categories in the polyphonic event detection task, we propose a disentangled feature (DF) to reduce interference among categories, which optimizes the high-level feature space by disentangling it based on class-wise identifiable information and obtaining multiple different subspaces. Experiments on the dataset of DCASE 2018 Task 4 show that the proposed SDS and DF significantly improve the detection performance of the embedding-level MIL approach with an attention pooling module and outperform the first place system in the challenge by 6.2 percentage points.

***Index Terms*—Sound event detection, machine learning, weakly-supervised learning, attention pooling.**

I. Introduction

SOUND event detection (SED) is the task to detect and recognize individual sound sources in realistic soundscapes. It is required to recognize not only the presence of each event category in a sound source but also the start and end boundaries of each existing event. Since sounds carry a large amount of information about our everyday environment, SED supports many applications in everyday life, such as noise monitoring for smart cities [1], bioacoustic species and migration monitoring [2], [3], surveillance [4], healthcare [5], and large-scale multimedia indexing [6].

For SED, annotations with detailed timestamps for all event occurrences are termed as strong annotations, while weak annotations only indicate the presence of event categories. The SED system learning with strong annotations carries out supervised learning while the SED system learning with only weak annotations carries out weakly-supervised learning. Due to the difficulty in obtaining large-scale strongly annotated training data, weakly supervised learning has become a new focus in research on SED, for which we mainly focus on weakly SED in this paper.

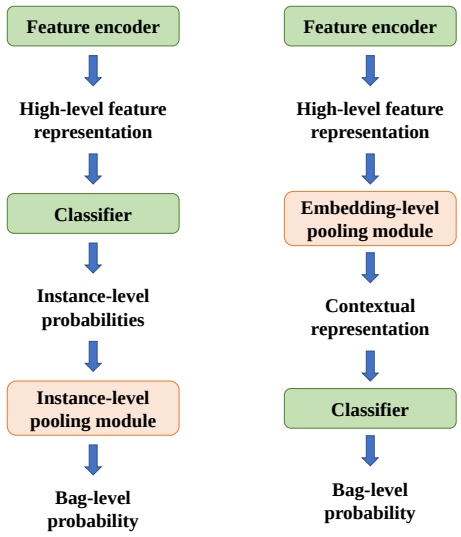
Weakly supervised learning is often approached as an MIL problem [7], [8]. It is especially common in medical image [9], [10] and semantic segmentation [11], [12]. The excellent performance of neural networks in various fields promotes the combination of the MIL framework and neural networks for weakly supervised learning [13], [14], [15], [16]. According to MIL, a bag of several instances has only bag-level annotations, in other words, does not have instance-level annotations. If there is at least one positive instance in a bag, the bag is annotated as a positive bag. Otherwise, the bag is annotated as a negative bag. The combination of the MIL framework and neural networks for SED focuses on how to integrate several frame-level outputs of neural networks into a clip-level output so as to enable the model to calculate loss with only clip-level annotations and carry out end-to-end learning. Since neural networks have been widely used as a general high-level feature extractor in various tasks, the MIL framework with neural networks typically comprises a neural network feature extractor which generates the high-level feature representation sequence and a pooling module such as global max pooling (GMP) [17], global average pooling (GAP) [18], global weighted rank pooling (GWRP) [19], [20], noisy-or pooling [21] and attention pooling [22], which integrates instance-level outputs into a bag-level output.

As mentioned in [22] and shown in Figure 1, the MIL approaches with neural network are distinguished as instance-level approaches and embedding-level approaches according to whether the pooling module involved in the MIL framework integrates instance-level probabilities into a bag-level probability or integrates instance-level high-level feature representations into a bag-level high-level feature representation.

Ilse et al. [22] claim that the embedding-level approach is superior to the instance-level approach in terms of bag-level classification and proposes an attention-based embedding-level approach, which performs the best compared to other MIL approaches. However, most researches in SED do not consider the embedding-level approach due to the fact that there are no instance-level probabilities involved in the procedure of the

[1]Bejing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

[2]University of Chinese Academy of Sciences, Beijing, China



(a) Instance-level approach (b) Embedding-level approach

Fig. 1. The comparison of the instance-level approach and the embedding-level approach.

embedding-level approach. Ilse et al. [22] mention that the attention weights in the proposed attention-based embedding-level approach are able to indicate key instances and provides an example of generating instance-level probabilities through post-processing (Max-min normalization) but takes no further investigation and does not carry out relevant experiments to prove its ability of instance-level classification. In fact, this simple method of taking post-processing result of attention weight as output probabilities, excessively depending on the distribution of maximum and minimum weights within a single bag, is likely to produce unstable instance-level predictions and lacks an adequate explanation. Therefore, in this paper, we provide a method for all the embedding-level approaches to generate instance-level probabilities. The proposed method considers that the instance-level high-level feature representations can share the same decision surface with the bag-level contextual representation, and we utilize a shared classifier in implementation. Furthermore, we explore the nature of the ability of the attention-based embedding-level approach to indicate key instances and thereby propose a method to generate more accurate instance-level probabilities for the attention-based embedding-level approach. Inspired by how the shared decision surface forms during training, the proposed method focuses on the forming of another latent specialized decision surface different from the shared decision surface during learning attention weight, based on which we re-design a classifier determined by this specialized decision surface and take probabilities output by this classifier as instance-level probabilities.

Besides weakly supervised learning, another problem of weakly-supervised SED we explore in this paper is how to carry out more efficient learning and more accurate detection for multi-category detection. Since multiple event categories tend to occur in co-occurrence in an audio clip, a SED system is also termed as a polyphonic SED system [23], [24], [25]. General polyphonic SED systems consider each event category equally. When designing models, they make all the categories share the same feature encoder. Hence, the high-level feature representations of all the categories are modeled into a same feature space. However, in realistic sound environment, multiple events overlapping in the unbalanced dataset, such as “Dishes” and “Frying”, would interference with the recognition of each other, especially when the numbers of clips of some event categories are relatively small and thereby less identifiable information about these event categories can be available. During training, the feature encoder tends to fit better for some event categories with more identifiable information than those with less identifiable information.

Therefore, by taking into account the category overlapping information, we propose a disentangled feature which re-models the high-level feature subspaces of the feature encoder to make the feature space of a certain category differ from those of the other categories without pre-training. We argue that if we allocate different high-level feature subspaces to different categories in advance according to the prior information, the mutual interference between categories will be reduced. In addition, if we relate the volume of the feature subspace of an event category to how much identifiable information about it can be available to train the model, the feature encoder can fit for all the event categories in a relatively balanced way.

A. *Our contributions*

In this paper, we approach SED as an MIL problem and utilize a neural network framework with pooling module to solve it.

We propose a shared decision surface for the embedding-level approach to generate instance-level probabilities. We argue that the classifier learning from the bag-level contextual representation forms a shared decision surface of both the bag-level contextual representation and the instance-level high-level feature representations, for which the instance-level probabilities could be obtained by passing the instance-level high-level feature representations through the same classifier directly.

Furthermore, we propose a specialized decision surface to make better instance-level predictions than the shared decision surface for the embedding-level attention pooling. We demonstrate that the embedding-level approach with the shared decision surface tends to perform better than the instance-level approach and the proposed specialized decision surface is more conducive to instance-level classification than the shared decision surface on the experimental dataset.

We also propose a disentangled feature, which re-models the high-level feature space so that the feature subspace of a certain category differs from other categories without pre-training, to improve the proposed weakly-supervised polyphonic SED system. The volume of these disentangled feature subspaces depend on the number of available clips containing strong category-wise identifiable information with less interference from other categories. In virtue of the introduction of more category-wise prior information as well as network redundancy weight reduction, the disentangled feature is able to improve the performance of polyphonic SED system.

Our experiments show that the embedding-level attention pooling module with specialized decision surface and disentangled feature outperforms other pooling modules as well as simple embedding-level attention pooling module. Detailed analysis of the high-level feature space in the experiment also supports our hypothesis.

The rest of this paper is organized as follows. We introduce related work about weakly-supervised polyphonic SED in Section II, describe in detail the MIL framework with different pooling modules in Section III, introduce the proposed methods in Section IV, describe the dataset and configuration of our experiments in Section V, analyze the results of experiments in Section VI and draw conclusions in Section VII.

II. Related work

A. Weakly-supervised SED

As mentioned in Section I, we approach weakly-supervised SED as an MIL problem. As for weakly-supervised SED, if we treat each frame in an audio clip as an instance, then the audio clip can be regarded as a bag with clip-level annotations (without frame-level annotations of frames). If a sound event occurs in any frame of the audio clip, the clip is considered as a positive clip of the event. Otherwise, the audio clip is treated as a negative audio clip.

Since a pooling module described in the previous section is essential to an MIL framework with neural network, there are lots of previous work about MIL with different pooling modules for weakly-supervised SED: a fully convolutional network with GMP [26], a joint detection-classification (JDC) model with an attention pooling module [27], a CNN based model with GAP [28], a gated CRNN with a softmax pooling module [29] and a joint separation-classification (JSC) model with a GWRP module [20].

Especially, McFee et al. [30] explore the effects of different pooling modules such as GAP, GMP and softmax pooling and propose an adaptive pooling module. Wang et al. [31] offer a comparison of several pooling modules including GAP, GMP, softmax pooling and attention pooling. However, all these work described above just considers the instance-level approaches. To explore the effects of different pooling modules cooperating with both of instance-level and embedding-level approaches on SED, we carry out a series experiments and find that the embedding-level approach tends to perform better. We describe in detail these effects in Section VI.

B. Polyphonic SED

SED is commonly grouped into two categories: monophonic and polyphonic SED. In monophonic SED, multiple sound events do not occur simultaneously, while in polyphonic SED, multiple events often occurs in co-occurrence. IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) is an influential challenge in this domain. The task 2 [32] of DCASE2016 is a monophonic SED task, while task 3 [33] of DCASE2016, task 4 [34] of DCASE2017 and task 4 [35] of DCASE2018 are polyphonic SED tasks. Obviously, polyphonic SED is closer to the realistic sound environment and more difficult to tackle. We maily focus on polyphonic SED in this paper.

Recently, neural networks such as recurrent neural network (RNN) [36] and convolutional recurrent neural network (CRNN) [25] show a significant effect on polyphonic SED. Commonly, these methods model each event category equally. However, in realistic sound environment, multiple events overlapping in the unbalanced dataset would interference with the recognition of each other, especially when the numbers of clips of some events are relatively small and less identifiable information about these events can be utilized in the model training. Imoto et al. [37] notice this and proposes a neural-network-based SED with graph Laplacian regularization based on the co-occurrence of sound events. We also focus on this problem and try to take advantage of more prior information about the data distribution to optimize the high-level feature space of the feature encoder mentioned in section I, thereby making more accurate classification for overlapping events.

III. MIL for weakly-supervised polyphonic SED

In this section, we describe in detail the MIL framework for weakly-supervised polyphonic SED. 8 common pooling modules including 4 instance-level pooling modules and 4 embedding-level pooling modules are introduced.

A. The MIL framework

For weakly-supervised polyphonic SED, since multiple different events might occur in co-occurrence in the same audio clip, we consider each event category separately when approaching SED as an MIL problem. Assuming that there are C event categories to detect, then for event category c, we treat an audio clip as a positive audio clip if the audio clip contains event category c. Otherwise, the audio clip is treated as a negative audio clip.

Let $\mathbf{x} = \{x_1, ..., x_T\}$ be the high-level feature representations of the audio clip generated by the feature encoder and $\mathbf{y} = \{y_1, ..., y_C\}$ ($y_c \in \{0, 1\}$) be the groundtruths, where C is the number of categories.

For the instance-level approach, the high-level feature representations are passed into the classifier to generate frame-level probabilities $\mathbf{P}(\mathbf{y} \mid x_t), ..., \mathbf{P}(\mathbf{y} \mid x_T)$.

Then the instance-level pooling module aggregates frame-level probabilities into a clip-level probability:

$$\hat{\mathbf{P}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{POOLING}\{\mathbf{P}(\mathbf{y} \mid x_1), ..., \mathbf{P}(\mathbf{y} \mid x_T)\} \quad (1)$$

When making predictions, assuming that α is a threshold for clip-level prediction and γ is a threshold for frame-level prediction. Then the clip-level prediction for event category c is:

$$\phi_{\mathbf{c}}(\mathbf{x}) = \begin{cases} 1, & \hat{\mathbf{P}}(1 \mid \mathbf{x}) \geq \alpha \\ 0, & otherwise \end{cases} \quad (2)$$

The the frame-level prediction for event c at time t is:

$$\varphi_{\mathbf{c}}(\mathbf{x}, t) = \begin{cases} 1, & \mathbf{P}(1 \mid x_t) \cdot \phi_{\mathbf{c}}(\mathbf{x}) \geq \gamma \\ 0, & otherwise \end{cases} \quad (3)$$

Without loss of generality, we set $\alpha = 0.5$ and $\gamma = 0.5$ in our experiments.

For the embedding-level approach, the embedding-level pooling module directly aggregates all the high-level feature representations into a contextual representation $\mathbf{h}$:

$$\mathbf{h} = \mathbf{POOLING}\left(x_1, ..., x_T\right) \tag{4}$$

Then the clip-level probability can be obtained by passing the contextual representation into the classifier:

$$\hat{\mathbf{P}}\left(\mathbf{y} \mid \mathbf{x}\right) = \mathbf{P}\left(\mathbf{y} \mid \mathbf{h}\right) \tag{5}$$

Therefore, the clip-level prediction for the embedding-level approach can be obtained according to Equation 2 and 5.

B. Instance-level pooling modules

We introduce 4 typical pooling modules for the instance-level MIL, namely global max pooling (GMP), global average pooling (GAP), global softmax pooling (GSP), and attention pooling (ATP). These 4 instance-level pooling modules are commonly used in weakly SED, such as GMP in [26], GAP in [28], GSP in [29], [38] and ATP in [27].

For GMP, the clip-level probability only depends on the maximum probability of all the frame-level probabilities of an audio clip:

$$\hat{\mathbf{P}}\left(\mathbf{y} \mid \mathbf{x}\right) = \max_t \mathbf{P}\left(\mathbf{y} \mid x_t\right) \tag{6}$$

For GAP, the clip-level probability relates to all the frame-level probabilities of an audio clip. More specifically, it takes the average value of all the frame-level probabilities as clip-level probabilities:

$$\hat{\mathbf{P}}\left(\mathbf{y} \mid \mathbf{x}\right) = \frac{1}{T}\sum_t \mathbf{P}\left(\mathbf{y} \mid x_t\right) \tag{7}$$

Obviously, since $\hat{\mathbf{P}}\left(\mathbf{y} \mid \mathbf{x}\right)$ only relates to the s^{th} high-level feature representation x_s ($s = \max_t \mathbf{P}\left(\mathbf{y} \mid x_t\right)$), GMP only updates a limited number of weights of the neural network for each clip. On the other hand, although GAP considers all the high-level feature representations when updating the neural network, it focuses on each frame equally, ignoring the different degree how much a frame contributes to the audio clip. Then a weighted pooling is proposed to fix this defect:

$$\hat{\mathbf{P}}\left(y_c \mid \mathbf{x}\right) = \sum_t a_{ct} \cdot \mathbf{P}\left(y_c \mid x_t\right) \tag{8}$$

where a_{ct} denotes the contribution of the t^{th} frame to an audio clip for event category c.

GSP and ATP are two examples of such weighted pooling modules, where GSP connects the contribution of frames with frame-level probabilities:

$$a_{ct} = \frac{\exp\left(\psi\left(\mathbf{P}\left(y_c \mid x_t\right)\right)\right)}{\sum_k \exp\left(\psi\left(\mathbf{P}\left(y_c \mid x_k\right)\right)\right)} \tag{9}$$

where ψ is a function to scale $\mathbf{P}\left(y_c \mid x_t\right)$ appropriately.

Different from GSP, ATP offers an independent detector to generate the contribution of frames. This independent detector is learnable and in this paper, we give a common form of such an independent detector:

$$a_{ct} = \frac{\exp\left((w_c^T x_t + b_c)/d\right)}{\sum_k \exp\left((w_c^T x_k + b_c)/d\right)} \tag{10}$$

where w_c^T and b_c are learnable parameters of the given independent detector and d is a scaling factor to avoid too-large value of $w_c^T x_t + b_c$. The value of d is generally consistent with the dimensions of x_t.

C. Embedding-level pooling modules

We describe how the 4 pooling modules discussed above cooperate with the embedding-level MIL approach. In fact, though the embedding-level MIL approach is introduced and claimed to be superior to the instance-level MIL approach in [22], its application in SED is rare.

Different from the instance-level pooling modules, the embedding-level pooling modules work by integrating high-level feature representations $\mathbf{x}$ instead of frame-level probabilities $\mathbf{P}\left(\mathbf{y} \mid x_t\right)$ as described in Section III-A.

For GMP, assuming that the contextual representation $\mathbf{h} = \{h_1, ..., h_e\}$ is an E dimensional vector and x_{te} is the e^{th} component of x_t, then the e^{th} component of the contextual representation $\mathbf{h}$ is

$$h_e = \max_t x_{te} \tag{11}$$

For GAP, the contextual representation $\mathbf{h}$ for event category c is:

$$\mathbf{h} = \frac{1}{T}\sum_t x_t \tag{12}$$

For GSP and ATP, the contextual representation h_c for event category c is:

$$h_c = \sum_t a_{ct} \cdot x_t \tag{13}$$

Similar to instance-level pooling modules, a_{ct}, the contribution of x_t to an audio clip for event category c is attained by Equation 9 for GSP and by Equation 10 for ATP.

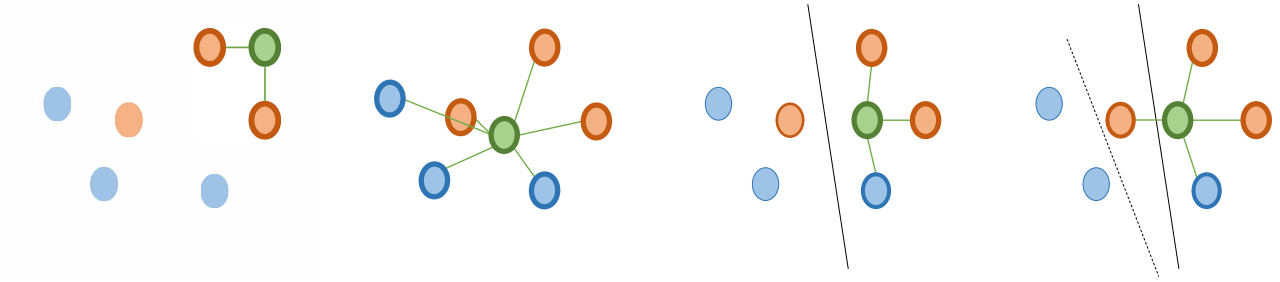
IV. Methods

In this section, we propose how to generate frame-level probabilities for the embedding-level approach and introduce the proposed specialized decision surface (SDS) and disentangled feature (DF).

A. Shared decision surface

Since there are no frame-level probabilities generated during learning, we propose that the clip-level contextual representation and frame-level high-level feature representations can share the same classifier, despite the fact that the classifier is simply utilized for classification of the contextual representation during training.

We argue that not only the model learns the decision surface (the classifier) explicitly for $\mathbf{h}$ but also learn a latent decision



(a) GMP (b) GAP (c) GSP (d) ATP

Fig. 2. A sketch of the relation of $\mathbf{h}$ and $\mathbf{x}$ of a positive audio clip in the 2-dimension feature space for a certain event category.

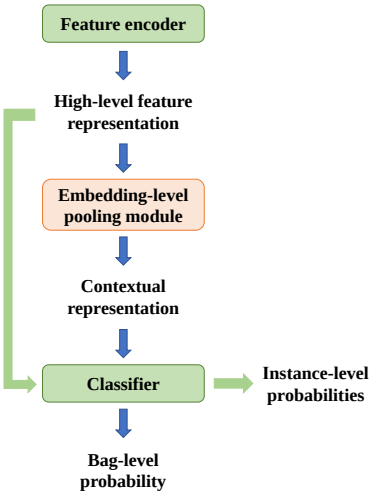


Fig. 3. The instance-level probabilities for the embedding-level approach.

surface for $\mathbf{x}$. Since this latent decision surface can not be obtained directly, we consider it to be close to the decision surface of $\mathbf{h}$.

As shown in Figure 2, to simplify the analysis, we assume that the high-level feature space is a 2-dimension space and sketch the relation of $\mathbf{h}$ and $\mathbf{x}$ in this 2-dimension feature space for a certain event category. The green circles represent $\mathbf{h}$, the orange circles represent positive frames in $\mathbf{x}$ and the blue circles represent negative frames in $\mathbf{x}$. We connect $\mathbf{h}$ and $\mathbf{x}$ in the following way: draw a line between $\mathbf{h}$ and a frame in $\mathbf{x}$ if there is a connection between them and let the thickness of the circle corresponding to the frame indicate the strength of the connection.

We assume that the decision surface is fixed, and explore how the feature encoder tends to form a high-level feature space to fit the decision surface. The relative position of the high-level feature representation of each frame to the decision surface changes constantly with the formation of the feature space. During training, the green circle of a positive audio clip tends to move toward the positive side of the decision surface, which implies that those circles connected with the green circles move with the green circle together. On the contrary, those circles connected with the green circle in a negative audio clip tends to move toward the negative side of the decision surface.

If circles (except green circles) with such a connection are considered to be positive and the strength of the connection is related to the confidence that it is considered positive, then the movement in a positive audio clip carries these (both true-positive and false-positive) circles toward the positive side of the decision surface. Since there is no true-positive circle in a negative audio clip, the movement in this circumstance actually carries false-positive circles toward the negative side of the decision surface. Attribute to these movements, positive (true-positive) frames and negative (false-positive) frames are able to gradually separate into two clusters and the decision surface of the contextual representations is close to the boundary of such two clusters, in other words, suitable for coarse frame-level classification, for which we regard it as a shared decision surface.

According to Equation 11, $\mathbf{h}$ in GMP only relates to two frames in an audio clip, the value of the high-level feature representation of which in one dimension is the largest of all frames, as shown in Figure 2(a). Thus the movements in GMP affect fewer frames in a negative audio clip but make fewer mistakes in a positive audio clip (carry fewer false-positive frames toward to the positive side of the decision surface). Similarly, $\mathbf{h}$ in GAP relates to all the frames and focuses on them equally as shown in Figure 2(b), for which it affects all the frames in a negative audio clip but makes more mistakes in a positive audio clip. For GSP, according to Equation 8 and 9, the strength of the connection a_{ct} depends on $\mathbf{P}(y_c \mid x_t)$ and $\mathbf{P}$ actually denotes the decision surface of $\mathbf{h}$. Therefore, since we consider that x_t and $\mathbf{h}$ share the same decision surface, the strength of the connection between x_t and $\mathbf{h}$ exactly depends on how much the model considers it as a positive frame. As shown in Figure 2(c), where the black solid line represents the decision surface mentioned above, if we ignore some of the relatively week connection, those green lines between $\mathbf{h}$ and x_t lying on the negative side of the decision surface can be neglected. Hence, GSP pursues a trade-off between affecting more frames in a negative audio clip and making fewer mistakes in a positive audio clip.

Therefore, as shown in Figure 3, we pass the frame-level high-level feature representation x_t through clip-level classifier to get frame-level probabilities $\mathbf{P}(y_c \mid x_t)$ for event c at time t. Then the the frame-level prediction is:

$$\varphi_{\mathbf{c}}(\mathbf{x}, t) = \begin{cases} 1, & \mathbf{P}(1 \mid x_t) \cdot \phi_{\mathbf{c}}(\mathbf{x}) \geq \gamma \\ 0, & otherwise \end{cases} \tag{14}$$

where $\mathbf{P}$ denotes the classifier of the contextual representation $\mathbf{h}$ according to Equation 5.

B. Specialized decision surface

To explore the more accurate boundary of the two clusters mentioned above, we propose a specialized decision surface (SDS). Different from the shared decision surface, SDS is not approximately close to but exactly the boundary of the two clusters so that SDS is able to provide more accurate frame-level detection.

Actually, for GMP and GAP, SDS does not present in an explicit way. Intuitively, they do not provide a explicit way to separate these two clusters. However, as for GSP, according to Equation 8 and 9, the forming of the two clusters depends on a_{ct} relating to the shared decision surface $\mathbf{P}$, so that SDS of GSP is coincident with the shared decision surface of GSP.

When it comes to ATP, according to Equation 10, the strength of the connection a_{ct} depends on the independent detector discussed in Section III-A instead of the shared decision surface $\mathbf{P}$ as shown in Figure 2(d). Therefore, free parameters w_c and b_c in the independent detector (dotted line in 2(d)) determine how to chose frames to move with the contextual representation together and gradually separate these frames from the rest. Although these movements try to promote the two clusters to distribute separately on opposite sides of the shared decision surface, the SDS utilized directly to select and separate the two clusters can be better matched the boundaries of the two clusters.

Therefore, the implement of SDS for the embedding-level attention pooling is based on a frame-level classifier: the combination of the independent detector utilized to generate a_{ct} with a activation layer employed to generate probabilities. Then the frame-level prediction for event c at time t is:

$$\varphi_{\mathbf{c}}(\mathbf{x}, t) = \begin{cases} 1, & p(1 \mid x_t) \cdot \phi_{\mathbf{c}}(\mathbf{x}) \geq \gamma \\ 0, & otherwise \end{cases} \tag{15}$$

$$p(1 \mid x_t) = \sigma\left(w_c^T x_t + b_c\right) \tag{16}$$

where w_c and b_c are free parameters of the independent detector and σ is an activation function to generate probabilities. We take Sigmoid as this activation function in our work.

C. Disentangled feature

For all the MIL approaches described above, the general feature encoder generates the high-level feature representations of all the categories from the same feature space. However, for multi-category classification, when a certain category often occur in co-occurrence with other categories, this approach makes it difficult to differentiate every single category. In other words, the forming of the high-level feature subspace of the event categories with insufficient identifiable information given in the training set will be largely disturbed by those categories occurring in co-occurrence with them. This effect will be exacerbated when the number of clips with much identifiable information of certain categories in the unbalanced set is particularly small.

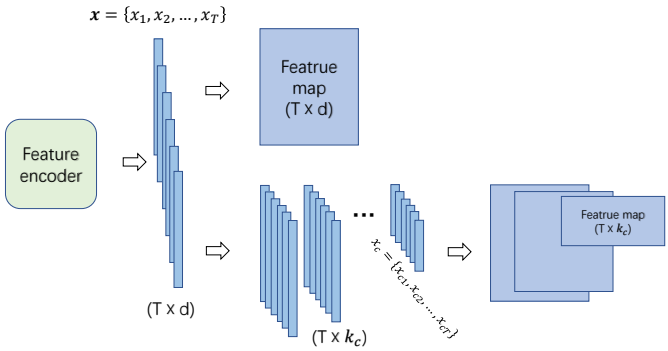


Fig. 4. The comparison of general feature and disentangled feature.

To mitigate this effect, we propose DF to re-model multiple feature subspaces for multiple categories. In this way, every category shares a different part of the feature encoder instead of the whole feature encoder and is allocated in advance a feature subspace of the high-level feature space generated by the feature encoder according to its priori information.

Assuming that $\chi^{\mathbf{d}}$ ($\mathbf{x} \in \chi^{\mathbf{d}}$) is a d-dimensional space generated by the feature encoder and $ß = \{e_1, e_2, ..., e_d\}$ is a basis of $\chi^{\mathbf{d}}$. We define χ_c, a subspace of $\chi^{\mathbf{d}}$, as the feature space of event category c. We produce χ_c by selecting specific bases of $\chi^{\mathbf{d}}$ and the basis of χ_c is

$$ß_c = \{e_1, e_2, ..., e_{k_c}\} \tag{17}$$

where $\mathbf{k} = \{k_1, k_2, ..., k_C\}$ $(0 < k_c \leqslant d)$ relates to the volume of χ_c.

In this way, the diversity of elements in $\mathbf{k}$ leads to the feature space of each category being remodeled into a disentangled feature space that is different from those of the other categories. For two categories i and j, the larger the absolute value of the difference between k_i and k_j is, the more different their feature space will be. The difference of feature spaces results in the diversity of decision surfaces among different categories without pre-training. In the extreme case with $k_1 = k_2 = ... = k_C = d$, all subspaces are equal to $\chi^{\mathbf{d}}$ so that the disentangled feature degenerates to general feature.

Meanwhile, we argue that the volume of χ_c is determined by the amount of available clips containing less interference. This is because that for category c, the larger the proportion of the clips containing less interference from other event categories is, the more the class-wise identifiable information needs to be learned, which requires the larger volume of the feature space. In contrast, the smaller the proportion of these clips is, the smaller volume of the feature space is required to prevent overfitting. For this reason, k_c increases as the proportion of these clips of category c increases.

Considering that too-small k_c severely cut into the ability of the model to recognize category c, we utilize a constant factor m $(0 \leqslant m \leqslant 1)$ to tackle this effect, then,

$$k_c = \lceil ((1 - m) \cdot f_c + m) \cdot d \rceil \tag{18}$$

where f_c $(0 \leqslant m \leqslant 1)$ relates to the number of clips containing less interference in the training set. As m increases to 1, disentangled feature degrades into general feature.

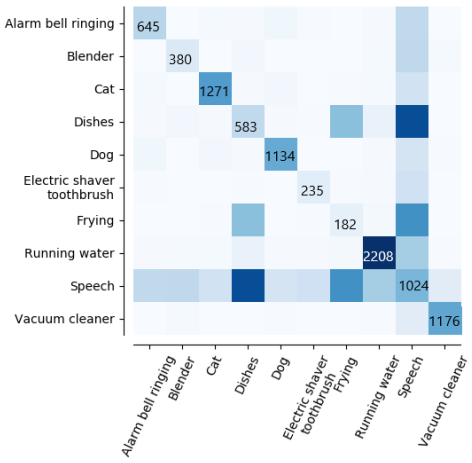


Fig. 5. The number of the clips where two categories occur in co-occurrence.

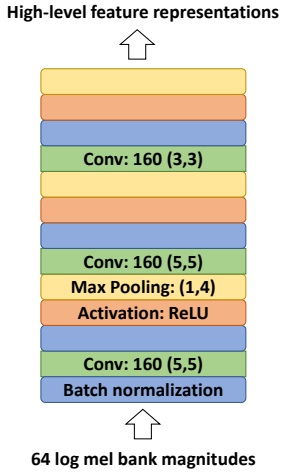


Fig. 6. The architecture of the feature encoder.

We quantify the level of interference according to the principle that the more categories a clip covers, the more interference the other categories cause to any one of them, then,

$$f_c = \sum_{i=1}^{C} \frac{r_i \cdot N_{ci}}{R} \tag{19}$$

$$R = \max_c \sum_{i=1}^{C} r_i \cdot N_{ci} \tag{20}$$

Here, N_{ci} denotes the number of clips containing i categories including category c in the training set and r_i is corresponding constant coefficient implying the importance of these clips. We argue that the less interference the other categories cause to any one of them in a clip, the more important the clip is, for which we determine r_i as:

$$r_i = \frac{1}{i} \, (1 \leqslant i \leqslant C) \tag{21}$$

We can also just consider those clips containing the least interference, then,

$$r_i = \begin{cases} 1, & i = 1 \\ 0, & otherwise \end{cases} \tag{22}$$

To simplify training, we take an orthogonal basis $ß^{'} = \{e_1, e_2, ..., e_d\}$ where the element of e_i in the i^{th} dimension is 1 for $\chi^{\mathbf{d}}$. Then the k_c basis vectors are related to k_c dimensions of x_t. As shown in Figure 4, we easily get a ladder-shape group of disentangled feature maps from feature encoder for a clip.

Combining disentangled feature $\mathbf{x_c} = \{x_{c1}, x_{c2}, ..., x_{cT}\}$ and the embedding-level attention module, to generate the contextual representation of event category c, we have

$$\mathbf{P}\left(y_c \mid \mathbf{x}\right) = \mathbf{P}\left(y_c \mid \mathbf{x_c}\right) = \mathbf{P}\left(y_c \mid h_c\right) \tag{23}$$

$$h_c = \sum_t a_{ct} \cdot x_{ct} \tag{24}$$

Then the contribution of x_{ct} to an audio clip is:

$$a_{ct} = \frac{\exp\left((w_c^T x_{ct} + b_c)/d_c\right)}{\sum_k \exp\left((w_c^T x_{ck} + b_c)/d_c\right)} \tag{25}$$

where w_c^T and b_c are learnable parameters mentioned in Section III-B and d_c is a scaling factor consistent with the dimensions of x_{ct}.

V. Experiments

In this section, we introduce the dataset and describe in detail the model architecture, the pre-processing, and post-processing methods, the training configuration, and the evaluation measure in our experiments.

A. Dataset

We utilize the dataset from task 4 of the DCASE 2018 Challenge [35], which is a subset of Audioset [39] by Google. The dataset consists of 10 categories of sound events from domestic environment: alarm/bell/ringing, blender, cat, dishes, dog, electric shaver/toothbrush, frying, running water, speech, and vacuum cleaner. The set contains 1578 weakly-labeled clips (2244 event occurrences) for which weak annotations have been verified and cross-checked, 14412 unlabeled in domain clips, 39999 unlabeled out-of-domain clips and 1168 clips with strong annotations. The challenge divides strong-labeled clips into two subsets: a validation set (288 clips) and an evaluation set (880 clips). In our experiments, we utilized the weakly labeled data to pre-train a clip-level classification model to tag unlabeled in domain data with weak annotations and wipe off 1001 clips with empty annotations. Consequently, the training set in our experiments embraces 14989 clips with noisy weak annotations, the characteristic of which are large scale and unbalanced distribution as shown in Figure 5.

B. Model architecture

The models employed in our experiments are divided into the instance-level model and the embedding-level model. As shown in Figure 1, both these two types of models comprise three modules: the feature encoder, the pooling module, and the classifier. The feature encoder is designed based on the model architecture of the baseline system of the task 4 [35]. We remove the RNN layer to make each frame-level high-level

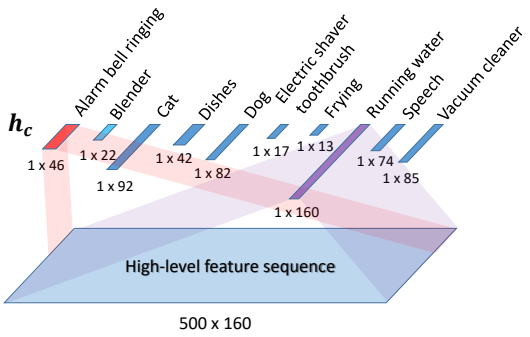


Fig. 7. A sketch of the high-level feature representations of the the embedding-level ATP-SDS with DF1.

TABLE I
THE DF DIMENSION AND THE WINDOW SIZE OF MEDIAN FILTERS WHEN $\beta = \frac{1}{3}$ PER CATEGORY.

Event	DF dimension		Window Size (frame)
	DF1	DFW	
Alarm bell ringing	46	31	17
Blender	22	22	42
Cat	92	43	17
Dishes	42	66	9
Dog	82	39	16
Electric shaver toothbrush	17	16	74
Frying	13	41	85
Running water	160	75	64
Speech	74	160	18
Vacuum cleaner	85	35	87

feature representation contain more identifiable information about the current frame, for which finer frame-level information is maintained. Meanwhile, the model thus more depends on the pooling module to integrate contextual information. We also remove dropout layers and increase the number of filters of CNN layers. The final feature encoder consists of 3 convolutional blocks, each of which comprises a convolutional layer, a batch normalization [40] layer, a max pooling layer (no temporal pooling), and an activation layer, as shown in Figure 6. The pooling modules including GAP, GMP, GSP and ATP are described in detail in Section III-B and Section III-C. We utilizes 1×1 convolutional layer with Sigmoid activation function as the classifier.

Different from other general pooling modules in the prediction phase, the instance-level and embedding-level ATP-SDS make frame-level prediction according to Equation 15 and Equation 16 discussed in Section IV-B.

As for DF, we experimented with two different methods for determination of the constant coefficient r_i discussed in Section IV-C: the embedding-level ATP-SDS with DFW (Equation 21) and the embedding-level ATP-SDS with DF1 (Equation 22). In addition, as mentioned in Section IV-C, since the hyper-parameter m is to avoid too-small k_c and for this dataset, each k_c is within a reasonable range, we set $m = 0$ in our experiments. Figure 7 illustrates the condition of the embedding-level ATP-SDS with DF1. More detailed information of disentangled dimessions for each category of the DFW and DF1 methods is shown in Table I.

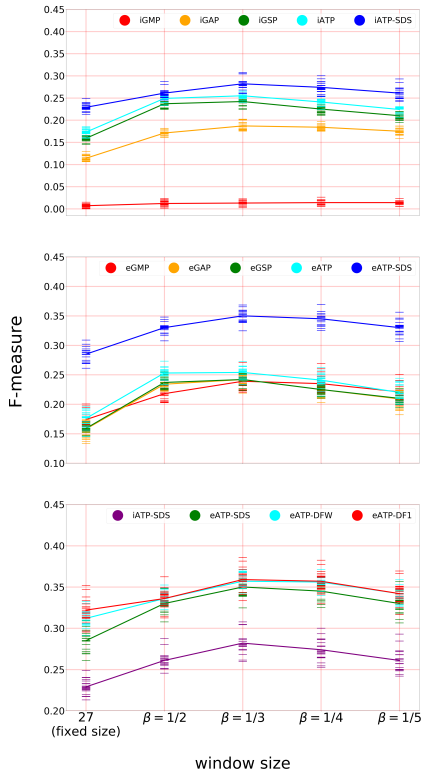


Fig. 8. The frame-level F_1 score of all 20 experiments of all the models with different window size of median filters. i* such as iGMP represents instance-level model and e* such as eGMP represents embedding-level model.

C. Pre-processing and post-processing

The feature passed into the feature encoder employed 64 log mel-bank magnitudes which are extracted from 40 ms frames with 50% overlap ($n_{FFT} = 2048$) using the librosa package [41]. All the 10-second audio clips are transformed to feature vectors with 500 frames. The threshold α (mentioned in Section III-A) of the predicted probability to determine whether an event category exists in a clip is 0.5. For frame-level prediction, all the probabilities are smoothed by a median filter with a group of adaptive window sizes. The operation of smoothing is repeated on the final frame-level prediction with a threshold $\gamma = 0.5$.

The adaptive window size of the median filter for category c is:

$$win_c = duration_c \cdot \beta \tag{26}$$

where $duration_c$ is the average duration of category c in the training set. In addition, we set $\beta = \frac{1}{3}$ and shows the specific window sizes in Table I.

D. Training and evaluation

The neural networks are trained using the Adam optimizer [42] with learning rate of 0.0018 and mini-batch of 64 10-second patches. The learning rate is reduced by 20% per 10 epochs. We take binary cross entropy as loss function. Training stops if there is no more improvement in clip-level macro

TABLE II
THE AVERAGE PERFORMANCE OF MODELS.

Model		Event detection (frame-level)			Audio tagging (clip-level)		
		$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$	**P**	**R**
Instance -level pooling	**GMP**	0.013 ± 0.010	0.089 ± 0.099	0.007 ± 0.007	0.565 ± 0.036	0.643 ± 0.070	0.531 ± 0.057
	GAP	0.187 ± 0.015	0.248 ± 0.023	0.172 ± 0.016	0.471 ± 0.012	0.682 ± 0.041	0.412 ± 0.024
	GSP	0.208 ± 0.032	0.266 ± 0.036	0.190 ± 0.035	0.487 ± 0.032	0.682 ± 0.041	0.412 ± 0.024
	ATP	0.255 ± 0.015	0.275 ± 0.020	0.257 ± 0.023	0.625 ± 0.033	$\mathbf{0.721 \pm 0.032}$	0.577 ± 0.051
	ATP-SDS	0.282 ± 0.026	0.322 ± 0.024	0.275 ± 0.037			
Embedding -level pooling	**GMP**	0.239 ± 0.032	0.249 ± 0.039	0.245 ± 0.041	0.626 ± 0.030	0.674 ± 0.030	0.610 ± 0.047
	GAP	0.242 ± 0.023	0.247 ± 0.030	0.265 ± 0.025	0.602 ± 0.019	0.657 ± 0.028	0.597 ± 0.039
	GSP	0.242 ± 0.018	0.243 ± 0.030	0.268 ± 0.022	0.608 ± 0.020	0.660 ± 0.028	0.603 ± 0.040
	ATP	0.254 ± 0.020	0.251 ± 0.023	0.297 ± 0.034	0.623 ± 0.033	0.667 ± 0.033	$\mathbf{0.634 \pm 0.058}$
	ATP-SDS	0.350 ± 0.025	0.367 ± 0.026	0.367 ± 0.034			
Embedding -level ATP-SDS	**DFW**	0.357 ± 0.016	0.363 ± 0.027	$\mathbf{0.373 \pm 0.018}$	$\mathbf{0.640 \pm 0.025}$	0.683 ± 0.038	0.622 ± 0.052
	DF1	$\mathbf{0.359 \pm 0.027}$	$\mathbf{0.378 \pm 0.035}$	0.371 ± 0.031	0.638 ± 0.026	0.688 ± 0.025	0.624 ± 0.047

TABLE III
THE BEST PERFORMANCE OF MODELS.

Model		Event detection			Audio tagging		
		$\mathbf{F_1}$	**P**	**R**	$\mathbf{F_1}$	**P**	**R**
Instance -level pooling	**GMP**	0.023	0.094	0.014	0.596	0.641	0.576
	GAP	0.202	0.271	0.186	0.482	0.670	0.410
	GSP	0.240	0.302	0.218	0.518	0.696	0.426
	ATP	0.268	0.278	0.280	0.653	**0.726**	0.620
	ATP-SDS	0.308	0.330	0.312			
Embedding -level pooling	**GMP**	0.271	0.274	0.286	0.648	0.683	0.639
	GAP	0.258	0.266	0.278	0.614	0.656	0.598
	GSP	0.252	0.245	0.279	0.628	0.667	0.616
	ATP	0.272	0.271	0.314	0.655	0.700	**0.640**
	ATP-SDS	0.369	0.393	0.370			
Embedding -level ATP-SDS	**DFW**	0.371	0.370	**0.380**	**0.656**	0.687	0.631
	DF1	**0.386**	**0.413**	0.377	0.652	0.680	**0.640**

F_1 performance on the validation set within 10 epochs. The best performing model on the validation set will be retained for prediction before the training stops. All the experiments are repeated 20 times under the same parameter configuration. We took the average of all the results as the final result. In particular, in order to compare with the performance of the first place in the challenge, we report the best results among these 20 experiments in addition. Event-based measures [43] with a 200ms collar on onsets and a 200ms / 20% of the events length collar on offsets are calculated over the entire test set. The implementation of our methods is available online at https://github.com/Kikyo-16/Sound_event_detection.

VI. DISCUSSION

In this section, we report the results of our experiments and analyze in detail the distribution of data in the high-level feature space of models to prove our conjecture.

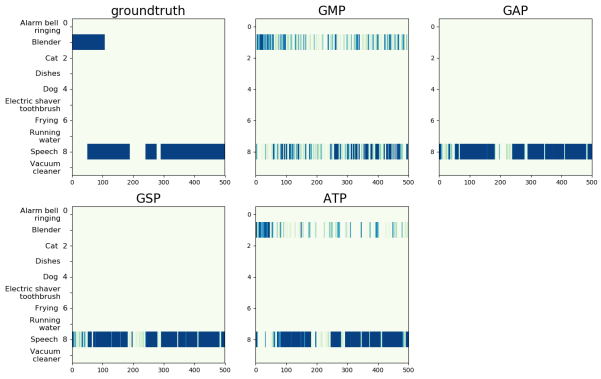


Fig. 9. Comparison of frame-level possibilities output by the classifier of the model with the groundtruth.

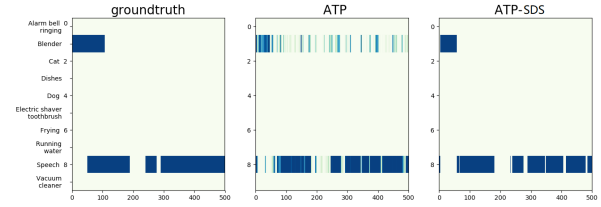


Fig. 10. Comparison of frame-level possibilities output by the classifier of the model with the groundtruth.

A. Results

We report the average results of 20 experiments of all the models in Table II and the best result among these 20 experiments of all the models in Table III. As shown in Table II, the embedding-level ATP-SDS with DF1 achieves the best average frame-level F_1 score of 0.359 among all the models. The best performance of ATP-SDS with DF1 shown in Table III achieves **0.386**, outperforming the first place (**0.324**) [44] in the challenge by **6.2** percentage points. The embedding-level ATP-SDS with DFW achieves the best average clip-level F_1 score of 0.640 among all the models and the best performance

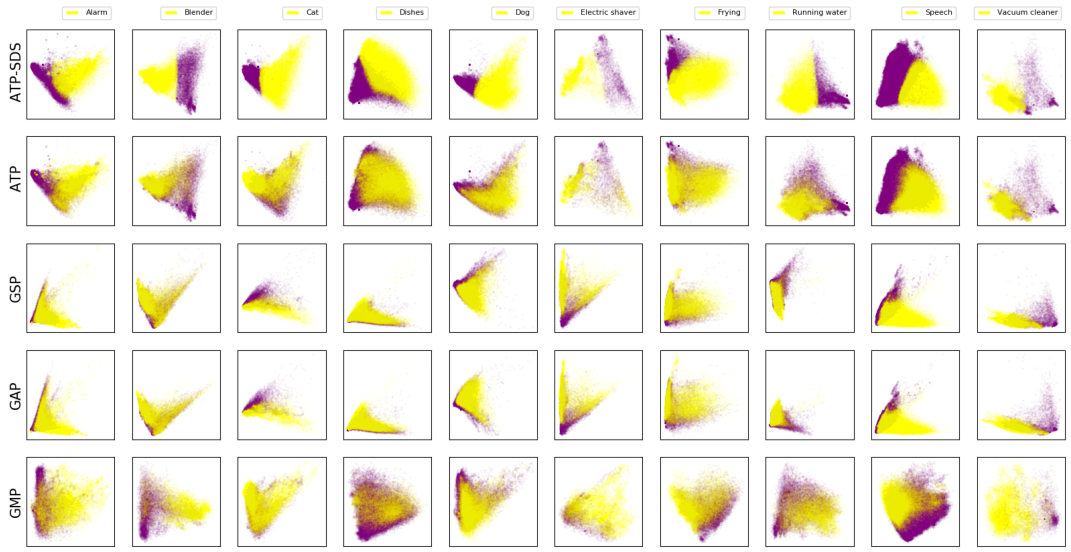


Fig. 11. The decision surfaces for different categories in the feature space generated from feature encoder (PCA).

of ATP-SDS with DFW is 0.656.

We illustrates all the results of the 20 experiments in Figure 8. As shown in Figure 8, a short horizontal line represents the result of one of the experiments and short horizontal lines distributing on the broken line represent the average results of 20 experiments of each model. Since different window sizes of median filters in post-processing have a great impact on results, we show frame-level performances of all the models when window sizes are fixed with 27 and adaptive window sizes employ different β ($\beta = \frac{1}{2}$, $\beta = \frac{1}{3}$, $\beta = \frac{1}{4}$ and $\beta = \frac{1}{5}$) respectively. As shown in in Figure 8, the value of β shows a significant effect on the frame-level performance of the model and the model tends to perform the best with $\beta = \frac{1}{3}$.

B. The performances of different pooling modules

By comparing the performances of the embedding-level models with those of the instance-level models in Table II, we find that the embedding-level models outperform the instance-level models, except that the average performances of the instance-level and the embedding-level ATP are almost the same. Without taking account of SDS and DF, ATP is dominant in event detection. We note that the embedding-level GMP performs best on audio tagging but worst on event detection. The best clip-level performance attributes to the fact that the strategy which GMP takes to select frames to update tends to make fewer mistakes. The relative poor frame-level performance is due to the fact that such a strategy which updates limited frames once a time leads to weak predictions of those frames that are not critical to the clip-level decision making.

As shown in Figure 9, we give an example audio clip of the test set to compare the frame-level performances of the 4 embedding-level models. Dark shadows in the figure represent frame-level probabilities output by the model without smoothing, the values of which range from 0 to 1. Compared with the groundtruth, GMP ignores those frames that are not critical to the clip-level decision making and make extremely discontinuous predictions both for "Blender" and "Speech", while the strategies that GAP and GSP take to select frames pay more attention to negative frames, leading to an incorrect prediction for "Blender" but achieving better boundary detection for "Speech". ATP achieves a better tradeoff between the two conditions above.

C. The effect of SDS on boundary detection

When we take SDS as decision surface for event detection, the frame-level average performance of the instance-level ATP is improved by 2.7 percentage points and that of the embedding-level ATP is improved by 9.6 percentage points as shown in Table II. As shown in Figure 10, we give an example audio clip of the test set to compare the frame-level performances of ATP and ATP-SDS. The predictions of ATP-SDS are obviously closer to groundtruths.

As shown in Figure 11, we transform the high-level feature representations generated from the feature encoders of the embedding-level models into a two-dimensional space using Principal components analysis (PCA) for observation. To highlight the frame-level decision surface, we only draw all the frames in the clips that are predicted to be positive. The yellow points represent frames predicted to be positive and purple points represent frames predicted to be negative. We can intuitively find that SDS clearly matches the boundary of two clusters discussed in Section IV-B. However, the shared decision surfaces of ATP, GSP, GAP and GMP show a poor ability to separate these two clusters and lead to poor performance on event detection. Among them, the constraint that the shared decision surface of GSP is exactly consistent with its SDS hinders the flexible formation of SDS and exerts

TABLE IV
THE AVERAGE FRAME-LEVEL CLASS-WISE PERFORMANCE OF MODELS (EVENT DETECTION).

Model		Alarm bell ringing	Blender	Cat	Dishes	Dog	Electric shaver toothbrush	Frying	Running water	Speech	Vacuum cleaner
Instance -level pooling	GMP	0.042	0.016	0.030	0.011	0.004	0.000	0.000	0.000	0.001	0.023
	GAP	0.280	0.087	0.065	0.046	0.102	0.240	0.362	0.123	0.176	0.389
	GSP	0.307	0.104	0.068	0.054	0.117	0.258	0.332	0.128	0.328	0.385
	ATP	0.351	0.191	0.053	0.125	0.173	0.392	0.332	0.170	0.381	0.385
	ATP-SDS	0.422	0.324	**0.246**	0.132	0.195	0.444	0.282	0.232	**0.480**	0.060
Embedding -level pooling	GMP	0.343	0.202	0.030	0.082	0.145	0.372	0.330	0.187	0.169	0.532
	GAP	0.316	0.069	0.045	0.067	0.123	0.362	0.344	0.210	0.442	0.443
	GSP	0.314	0.071	0.051	0.080	0.132	0.350	0.327	0.210	0.449	0.436
	ATP	0.328	0.144	0.043	0.088	0.171	0.329	0.333	0.192	0.467	0.447
	ATP-SDS	0.475	0.327	0.234	0.128	0.195	0.478	0.339	0.253	**0.480**	**0.587**
Embedding -level ATP-SDS	DFW	0.444	0.301	0.232	**0.159**	**0.230**	0.454	**0.467**	**0.270**	0.456	0.560
	DF1	**0.476**	**0.365**	0.233	0.151	0.216	**0.512**	0.457	0.240	0.419	0.524

TABLE V
THE AVERAGE CLIP-LEVEL CLASS-WISE PERFORMANCE OF MODELS (AUDIO TAGGING).

Model		Alarm bell ringing	Blender	Cat	Dishes	Dog	Electric shaver toothbrush	Frying	Running water	Speech	Vacuum cleaner
Instance -level pooling	GMP	0.729	0.420	0.486	0.405	0.537	0.531	0.492	**0.564**	0.857	0.633
	GAP	0.511	0.223	0.525	0.463	0.474	0.401	0.550	0.254	0.720	0.594
	GSP	0.534	0.258	0.530	0.446	0.469	0.442	0.551	0.281	0.741	0.613
	ATP	0.740	0.547	**0.646**	0.573	0.551	**0.655**	0.554	0.513	0.870	0.604
	ATP-SDS										
Embedding -level pooling	GMP	0.722	0.535	0.607	0.555	0.583	0.584	0.578	0.562	**0.874**	0.657
	GAP	0.740	0.390	0.577	0.547	0.584	0.605	0.588	0.511	0.837	0.639
	GSP	0.753	0.396	0.582	0.554	0.589	0.583	0.595	0.535	0.835	0.660
	ATP	0.749	0.505	0.614	0.552	0.614	0.580	0.550	0.517	0.870	**0.679**
	ATP-SDS										
Embedding -level ATP-SDS	DFW	**0.775**	0.563	0.608	**0.574**	**0.62**4	0.573	**0.602**	0.548	0.868	0.666
	DF1	0.762	**0.565**	0.614	0.554	0.623	0.612	0.583	0.547	0.864	0.655

a negative effect on the separation of the two clusters. These observation exactly meets what we expect in Section IV-B.

D. The effect of DF on multi-class classification

When we combine embedding-level ATP-SDS with DF, the frame-level average performance of ATP-SDS with DFW outperforms ATP-SDS by 0.7 percentage points and that of ATP-SDS with DF1 outperforms ATP-SDS by 0.9 percentage points as shown in Table II. We report average frame-level F_1 and average clip-level F_1 of 20 experiments of all 10 event categories in Table IV and Table V.

As shown in Figure 5, "Dishes" and "Frying" often occur in co-occurrence with each other and have a relatively small proportion in the training set while "Speech" often occurs in co-occurrence with any other categories. "Running water" also often occurs in co-occurrence with "Dishes". As shown in Table IV and Table V, ATP-SDS with DFW and DF1 did improve class-wise performances of "Dishes", "Frying" and "Running water" both on event detection and audio tagging. This is because the separate subspaces of each category reduce the interference between them. However, the class-wise performance of ATP-SDS with DFW and DF1 "Speech" is a little worse. We argue that since the number of clips containing "Speech" is much larger than that of clips containing any other event categories in the training set, the feature encoder without employing DF tends to form a high-level feature space more suitable for the event category "Speech". When DF is employed, the other categories are much less disturbed by "Speech" and make a more balanced contribution to the feature encoder, which raises overall class-wise performances but also leads a little poorer performance of category "Speech".

VII. CONCLUSIONS

In this paper, we present how to generate frame-level probabilities for the embedding-level MIL approach and propose a specialized decision surface (SDS) and a disentangled feature (DF) for weakly-supervised polyphonic SED.

Firstly, we approach it as an MIL problem and then introduce an MIL framework with neural networks and pooling module. This framework is common in some weakly-supervised tasks and is grouped into two approaches: the instance-level approach and the embedding-level approaches. We enable the embedding-level approach to make instance-level predictions and demonstrate the embedding-level approach tends to outperform the instance-level approach on experimental dataset. Based on the exploration of the ability of the embedding-level approach to produce frame-level probabilities, we propose a specialized decision surface (SDS) to detect more accurate boundaries of events.

Secondly, to tackle the common problem causing by category co-occurrence between categories and data unbalance in the multi-label task, we propose a disentangled feature, which determines several certain subspaces for different categories without pre-training according to the prior information. In terms

of optimizing the structure of the neural network, DF reduces redundant weights in the network and improves the training efficiency. From the perspective of feature space, DF optimizes the feature encoder and reduces the volume of high-level feature space of categories with insufficient samples, thus making it easier to learn more compact distribution. At the same time, DF, combined with prior information about co-occurrence between categories, reduces the interference between categories and improves the performance of the model.

Finally, the results of experiments on the dataset of DCASE2018 task 4 confirm our conjecture, which achieves a frame-level F_1 of 38.6%, outperforming the first place in the challenge by 6.2 percentage points.

ACKNOWLEDGMENT

This work is partly supported by Beijing Natural Science Foundation (4172058).

REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. DuBois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for the monitoring, analysis and mitigation of urban noise pollution," *arXiv preprint arXiv:1805.00889*, 2018.

[2] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[3] J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck, and S. Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," *PloS one*, vol. 11, no. 11, p. e0166866, 2016.

[4] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, p. 52, 2016.

[5] S. Goetze, J. Schroder, S. Gerlach, D. Hollosi, J.-E. Appell, and F. Wallhoff, "Acoustic monitoring and localization for social care," *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, 2012.

[6] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[7] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in neural information processing systems*, 1998, pp. 570–576.

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[9] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, "Multiple-instance learning for medical image and video analysis," *IEEE reviews in biomedical engineering*, vol. 10, pp. 213–234, 2017.

[10] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, I. Eric, and C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 1626–1630.

[11] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1742–1750.

[12] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu, "Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 256–263.

[13] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," *arXiv preprint arXiv:1412.7144*, 2014.

[14] Z.-H. Zhou and M.-L. Zhang, "Neural networks for multi-instance learning," in *Proceedings of the International Conference on Intelligent Information Technology, Beijing, China*, 2002, pp. 455–459.

[15] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3460–3469.

[16] O. Z. Kraus, J. L. Ba, and B. J. Frey, "Classifying and segmenting microscopy images with deep multiple instance learning," *Bioinformatics*, vol. 32, no. 12, pp. i52–i59, 2016.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 685–694.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[19] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 695–711.

[20] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "A joint separation-classification model for sound event detection of weakly labelled data," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 321–325.

[21] Y. Wang, J. Li, and F. Metze, "Comparing the max and noisy-or pooling functions in multiple instance learning for weakly supervised sequence learning tasks," *Proc. Interspeech 2018*, pp. 1339–1343, 2018.

[22] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," *arXiv preprint arXiv:1802.04712*, 2018.

[23] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–7.

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

[25] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[26] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 791–795.

[27] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "A joint detection-classification model for audio tagging of weakly labelled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 641–645.

[28] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 326–330.

[29] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[30] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 11, pp. 2180–2193, 2018.

[31] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[32] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 85–92.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[34] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and

human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.
[35] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. Parag Shah, "Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments," July 2018, submitted to DCASE2018 Workshop. [Online]. Available: https://hal.inria.fr/hal-01850270
[36] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, T. Hayashi, S. Watanabe, T. Toda, T. Hori *et al.*, "Duration-controlled lstm for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 11, pp. 2059–2070, 2017.
[37] K. Imoto and S. Kyochi, "Sound event detection using graph laplacian regularization based on event co-occurrence," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1–5.
[38] J. Yan, Y. Song, W. Guo, L.-R. Dai, I. McLoughlin, and L. Chen, "A region based attention method for weakly supervised sound event detection and classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 755–759.
[39] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.
[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[41] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 24.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[43] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162
[44] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

Liwei Lin received the B.E. degree in Computer Science and Technology from China Agricultural University, Beijing, China, in 2017. She is currently pursuing an M.E. degree in Computer Science and Technology in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. her research interest includes audio signal processing and machine learning.

Xiangdong Wang is an associate professor in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He received Doctors degree in Computer Science at Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007. His research field includes human-computer interaction, speech recognition and audio processing.

Hong Liu is an associate professor in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. She received her Doctors degree in Computer Science at Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007. Her research field includes human-computer interaction, multimedia technology, and video processing.

Yueliang Qian is a professor in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He received his Bachelors degree in Computer Science at Fudan University, Shanghai, China in 1983. His research field includes human-computer interaction and pervasive computing.