INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Few-shot learning of new sound classes for target sound extraction

Marc Delcroix, Jorge Bennasar Vázquez, Tsubasa Ochiai, Keisuke Kinoshita, Shoko Araki

NTT Corporation, Japan
marc.delcroix@ieee.org

Abstract

Target sound extraction consists of extracting the sound of a target acoustic event (AE) class from a mixture of AE sounds. It can be realized using a neural network that extracts the target sound conditioned on a 1-hot vector that represents the desired AE class. With this approach, embedding vectors associated with the AE classes are directly optimized for the extraction of sound classes seen during training. However, it is not easy to extend this framework to new AE classes, i.e. unseen during training. Recently, speech, music, or AE sound extraction based on enrollment audio of the desired sound offers the potential of extracting any target sound in a mixture given only a short audio signal of a similar sound. In this work, we propose combining 1-hot- and enrollment-based target sound extraction, allowing optimal performance for seen AE classes and simple extension to new classes. In experiments with synthesized sound mixtures generated with the Freesound Dataset (FSD) datasets, we demonstrate the benefit of the combined framework for both seen and new AE classes. Besides, we also propose adapting the embedding vectors obtained from a few enrollment audio samples (few-shot) to further improve performance on new classes.
Index Terms: Sound extraction, Adaptation, Acoustic event, Deep learning

1. Introduction

Human beings can focus on listening to a desired sound in a mixture, which enables us to follow our interlocutor despite others' conversations, to carefully listen to the violin part in a concert, or to pick up important acoustic events (AE) sounds such as a klaxon at a crossing. It has been a long-standing goal of researchers to reproduce human listening capabilities. Recently, neural network-based target sound extraction has received increased interest as a promising approach towards this goal, with methods developed to extract speech of a target speaker [1–4], music instruments [5, 6] or AE sounds [7–9]. In this paper, we focus on the AE sound extraction problem, which is particularly challenging given the large variety of sounds it covers (e.g. knock, telephone rings, cough, animal sounds, etc).

AE sound extraction consists of extracting the sound of a target AE class from a mixture of AE sounds. It uses an extraction neural network that estimates the target AE sound given the sound mixture and an embedding vector that represents the characteristics of the target sound. The embedding vector can be obtained using an embedding encoder that receives either (1) an *enrollment audio sample* that is similar to the target AE sound [8] or, (2) a *1-hot vector* that represents the target AE class [7, 9]. The extraction neural network and the embedding encoders are jointly trained.

The enrollment-based approach does not explicitly assume well-defined AE classes and thus can naturally handle target sounds from *new classes* (i.e. unseen during training), given a short enrollment audio sample of a sound of that new class. However, the embedding vectors may not be optimal for certain AE classes if the AE class has relatively large intra-class variability e.g. animal sounds or phone rings, which may cause a mismatch between the enrollment and the target sound in the mixture. In contrast, the 1-hot vector provides a way to optimize directly the embedding for the sound classes during training. It can thus achieve optimal performance on the seen AE classes. However, it cannot handle new AE classes.

In this paper, we propose a method to combine the advantages of both approaches by designing a *mixed model* that can perform both 1-hot- and enrollment-based target sound extraction. Since both approaches use embedding vectors, by mapping these embedding vectors to a common space, we can use either 1-hot- or enrollment-based approaches to extract a target sound. This is realized by training with a multi-task scheme, i.e. training a shared extraction network simultaneously with embedding vectors obtained from 1-hot vectors and enrollment samples. With such a mixed model, we can directly optimize the embedding vectors for seen classes with the 1-hot encoder, while enabling the handling of new AE classes with the enrollment encoder. Besides, the regularization effect of the multi-task training scheme can also help to learn better models.

In experiments using mixtures generated from sound events taken from the Freesound Datasets (FSDs) [10, 11], we confirm that the proposed mixed model can achieve superior performance on the extraction of seen AE classes compared to the baseline 1-hot- or enrollment-based approaches. Moreover, it can also generalize to new classes as the enrollment-based approach. We further investigate an adaptation/retraining strategy that enables to boost performance on new AE classes given a few samples from the new AE classes. The proposed method can thus learn 1-hot-based target sound extraction with a few labeled data of the new AE classes, which can be seen as few-shot learning.

We base our investigation using a network configuration similar to that of SpeakerBeam [12], which we proposed for enrollment-based target speech extraction. We thus refer to the target sound extraction framework as SoundBeam. In the remainder of the paper, we first describe the SoundBeam framework and the proposed mixed model in Section 2. In Section 3, we discuss how to handle new classes and introduce the adaptation framework. We introduce related works in Section 4. In Section 5, we present experimental results on seen and new AE classes. Finally, we conclude the paper in Section 6.

2. SoundBeam

We consider the problem of extracting sounds of a target AE class from a mixture of AE sounds given as,

$$\mathbf{y} = \mathbf{x} + \mathbf{i}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^T$ and $\mathbf{x} \in \mathbb{R}^T$ are the observed single-channel mixture and the target sound signals, respectively. $\mathbf{i} \in \mathbb{R}^T$ is a signal that includes the interference signals from other AE

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-1369

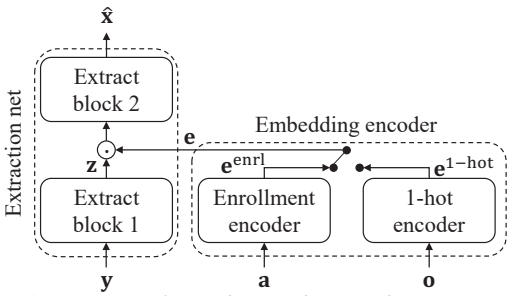

Figure 1: *Diagram of SoundBeam showing the two ways to obtain the embedding vectors, i.e. 1-hot- or enrollment-based.*

classes and background noise, and T is the signal duration.

Figure 1 is a schematic diagram of SoundBeam. It consists of two networks (1) an *extraction network* and (2) an *embedding encoder* network. The extraction network estimates the target sound signal, $\hat{\mathbf{x}} \in \mathbb{R}^T$, as,

$$\hat{\mathbf{x}} = f(\mathbf{y}, \mathbf{e}), \tag{2}$$

where $f(\cdot)$ is a neural network, $\mathbf{e} \in \mathbb{R}^{D\times 1}$ is an embedding vector that represents the characteristics associated with the target AE class, and D is the dimension of the embedding vector.

There are different ways to exploit the embedding vector within the extraction network [2,8,13]. We use here an element-wise multiplication ($\odot$ operator in Fig. 1) between the internal representation of the mixture signal after the first extraction block, $\mathbf{z}$, and the embedding vector, $\mathbf{e}$. Note that the extraction network could directly predict the target sound signal or a mask that is applied to the mixture to extract the target sound signal as in [12]. We use the mask-based approach in this study.

The embedding encoder computes the embedding vector $\mathbf{e}$. There are two ways of estimating the embedding vector: using a 1-hot vector [7, 9] or an enrollment audio sample [8]. We describe these two approaches in the following subsections.

2.1. 1-hot-based SoundBeam

Let $\mathbf{o} = [o_1, \ldots, o_N]^T$ be a 1-hot vector, which characterizes the AE class of the target sound, i.e. a $N \times 1$ vector with i-th element given as $o_i = \delta_{i,n}$, where $\delta_{i,n}$ is the Kronecker delta, n is the index of the target AE class, and N is the number of AE classes. The 1-hot encoder consists of an embedding layer. The embedding vector obtained from the 1-hot vector, $\mathbf{e}^{\text{1-hot}} \in \mathbb{R}^{D\times 1}$, is given by,

$$\mathbf{e}^{\text{1-hot}} = \mathbf{W}\mathbf{o}, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{D\times N}$ is an embedding matrix, whose columns contain the embedding vectors for each AE class.

The embedding layer can be trained jointly with the extraction network [7, 9], which enables to directly optimize the embedding vectors $\mathbf{e}^{\text{1-hot}}$ for each AE sound class. As a side-effect, the AE sound classes that the method can handle are fixed by the embedding matrix $\mathbf{W}$ and are limited to the AE sound classes seen during training. After training, it is thus not straightforward to handle new AE classes.

2.2. Enrollment-based SoundBeam

Another approach to obtain the embedding vector is to use an enrollment audio sample, $\mathbf{a} \in \mathbb{R}^{T^a}$, of a sound similar to the target sound. Note that in general the duration of the enrollment signal T^a may differ from the duration of the mixture T. The embedding derived from the enrollment sample, $\mathbf{e}^{\text{enrl}} \in \mathbb{R}^{D\times 1}$, can be obtained as,

$$\mathbf{e}^{\text{enrl}} = g(\mathbf{a}), \tag{4}$$

where $g(\cdot)$ represents the enrollment encoder, which is a neural network that maps the enrollment signal to a $D \times 1$ vector using e.g. average pooling across time-axis.

The embedding-based approach does not directly optimize the embedding vector for the AE classes, which can result in lower performance for seen AE classes. However, the method can naturally handle new AE classes, when we provide an enrollment sample with similar sound characteristics as the target, and if the system has been trained with a sufficient variety of AE sounds [8].

2.3. Training a shared embedding space: SoundBeam-mixed

The above formulation of target sound extraction emphasizes the similarities of the 1-hot- and enrollment-based approaches. Both approaches represent AE sounds in an embedding space. However, when trained separately, the 1-hot- and enrollment-based approaches will lead to different representations of the AE sounds. We propose to map the embedding vectors obtained by both approaches to a common embedding space so that we can enjoy the advantages of both frameworks, i.e. optimal performance on seen AE classes and generalization to new AE classes.

We realize this by creating a model (called *SoundBeam-mixed*) that includes both the enrollment and the 1-hot encoders and a shared extraction network. SoundBeam-mixed is trained with multi-task (or multi-branch) training, i.e. we alternate between the 1-hot and the enrollment encoders during training. This will ensure that both the 1-hot and enrollment encoders provide a similar representation of AE sounds. Besides combining the strength of both approaches, we also expect improved extraction performance with SoundBeam-mixed thanks to the regularization effect of multi-task learning.

The training loss of *SoundBeam-mixed* consists of,

$$\begin{aligned}\mathcal{L} =& \mathcal{L}^{ext}\left(\hat{\mathbf{x}} = f(\mathbf{y}, \mathbf{e}^{\text{enrl}}), \mathbf{x}\right) + \mathcal{L}^{ext}\left(\hat{\mathbf{x}} = f(\mathbf{y}, \mathbf{e}^{\text{1-hot}}), \mathbf{x}\right) \\ &+ \alpha\mathcal{L}^{emb}(\mathbf{e}^{\text{enrl}}, \mathbf{e}^{\text{1-hot}}),\end{aligned} \tag{5}$$

where $\mathcal{L}^{ext}(\cdot)$, $\mathcal{L}^{emb}(\cdot)$ and α are an extraction loss, an embedding loss (EL) and a weighting parameter, respectively. Note that the other versions of SoundBeam are trained using only an extraction loss. We use the negative scale-dependent signal-to-noise ratio (SNR) [14] as the extraction loss,

$$\mathcal{L}^{ext}(\hat{\mathbf{x}}, \mathbf{x}) = -10\log_{10}\left(\frac{\|\mathbf{x}\|^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}\right) \tag{6}$$

Various EL could be used. Here we simply compute the distance between the embedding vectors of both encoders as,

$$\mathcal{L}^{emb} = d(\mathbf{e}^{\text{enrl}}, \mathbf{e}^{\text{1-hot}}), \tag{7}$$

where $d(\cdot)$ is the cosine distance between the embedding vectors.

Note that by simply sharing the extraction network parameters, we already constrain both encoders to map their inputs into a common embedding space. However, the additional embedding loss (EL) may further help the embedding spaces becoming more similar. The multi-task loss proposed in Eq.(5) may bring a mutual regularization effect, i.e. the 1-hot encoder may help the enrollment encoder learn more general sound embedding vectors, while the enrollment encoder may help the 1-hot encoder capture sound characteristics of the AE classes.

3. Handling new classes

For practical applications, it is important to be able to flexibly extend a sound extraction system to new AE classes after deployment using only a few audio samples from the new classes. Let us assume that we have K enrollment audio samples $\{\mathbf{a}_1, \ldots, \mathbf{a}_K\}$ from a new AE class. Thanks to the enrollment encoder, enrollment-based SoundBeam or mixed SoundBeam can naturally handle new classes.

For example, to extract AE sounds from a new class, we can average the K embedding vectors computed with the enrollment encoder as,

$$\mathbf{e}^{new} = \frac{1}{K} \sum_{k=1}^{K} g(\mathbf{a}_k). \tag{8}$$

We can then use $\mathbf{e}^{new}$ as the embedding vector for the new AE class.

To further improve the quality of the embedding vectors, we propose to directly adapt the new embedding matrix vectors. To do that, we first register the new classes by adding $\mathbf{e}^{new}$ to the embedding matrix as,

$$\mathbf{W}' = [\mathbf{W}, \mathbf{e}^{new}], \tag{9}$$

where $\mathbf{W}' \in \mathbb{R}^{D \times (N+1)}$ is a new embedding matrix. We then retrain the embedding vector of the new class by using adaptation data. Here, we create an adaptation dataset by mixing sounds from the training data and the K enrollment audio samples. Note that, if we fix all network parameters but the new embedding, $\mathbf{e}^{new}$, we can adapt the embedding vector for the new AE class while keeping the previous embedding vectors unchanged, which will guarantee a constant level of performance for the AE classes seen during training.

4. Related works

Target sound extraction can be realized by combining sound separation [15, 16] and event identification [17–20]. However, separation approaches require knowing or estimating the number of sources. Moreover, a single model as SoundBeam can be jointly optimized, which can lead to superior performance [7].

The 1-hot-based SoundBeam described in section 2.1 corresponds to the universal sound extractor [7]. Concurrently [9] proposed a similar 1-hot-based sound extraction system that combines a sound event detection with an extraction network, to allow training on weakly labeled data consisting of audio clips containing several AEs without labels for their time occurrence.

The enrollment-based SoundBeam follows a similar implementation as SpeakerBeam [12], but is applied to AE sounds instead of speech. It is also conceptually similar to [6, 8] but with different network architectures and different training schemes. Both works assumed AE class labels were not available during training and assured similar sound characteristics for the target enrollment by using either a different portion of the same sound sample [8], or the same sample for enrollment and target sound during training [6]. Moreover, [6] proposed converting the model to a 1-hot-based model using averaging of the embedding of the same class in the training set when labels were available, similar to Eq. (8). However, this approach does not directly optimize the embedding vectors as our proposed SoundBeam-mixed does. Besides, it was tested only for music sound extraction, with a small number of seen classes.

Compared to [6, 8, 9], the proposed SoundBeam-mixed requires labeled data to train the 1-hot encoder. However, we could exploit both labeled and unlabelled data by borrowing the idea of [8] to augment the data to train the enrollment encoder. Besides, we could also use a similar approach to [9, 21] to generate labels to train the 1-hot encoder with weakly-labeled training data. These will be part of our future works.

5. Experiments

5.1. Dataset

To evaluate the effectiveness of the proposed method, we created a dataset of simulated sound event mixtures based on the FSD-Kaggle 2018 [10] and the FSD50K corpora [11].

We created mixtures by mixing 3 AE sounds randomly selected from different AE classes. We included stationary background noise to the mixtures at a signal-to-noise ratio (SNR) between 15 and 25 dB, using noise samples from the REVERB challenge corpus (REVERB) [22]. All mixtures are six-second long and were generated by randomly extracting three audio clips of 2 to 5 seconds from the FSD corpus and pasting (adding) them to random time-positions on top of the six-second background noise. We created sound event mixtures by utilizing Scaper [23]. In this experiment, we downsampled the sounds to 8 kHz to reduce the computational and memory costs. All experiments were performed with mixtures of 3 AE classes, but We confirmed in [7] that SoundBeam could handle mixtures with more AE classes.

The training and development sets consist of 50,000 and 10,000 mixtures, respectively. We used sound samples randomly selected from the 41 AE classes from the training set of the FSD-Kaggle dataset. These include AEs such as human sounds, object sounds, musical instruments, etc [10]. We generated two test sets using the FSD-Kaggle and a subset of the FSD50K data that consists of the AE sound samples from a single AE class provided in the FUSS dataset [24]. We used FSD50K to generate data with new AE classes unseen in the FSD-Kaggle training set. The first test set (*mixtures of seen AE classes*) consists of 10000 mixtures of sounds from the AE classes seen during training generated from FSD-Kaggle and is the same as that used in [7]. The second test set (*mixtures with new AE classes*) consists of 3000 mixtures of sounds from two seen classes and one new AE class out of 10 new AE classes (including male speech, electric guitar, camera, etc). It is generated by using both FSD-Kaggle and FSD50K.

For the enrollment-based experiments on seen classes, we randomly selected an enrollment audio sample from the AE class of the target AE sound that differs from the target sound. For the experiment on new classes, we randomly selected K enrollment utterances to compute the average embedding vectors as shown in Eq. (8). We performed experiments with K=1, 5, 10. We also created an adaptation set, which consisted of 1000 mixtures generated by mixing sounds from 2 AE classes from the training set and 1 sound from the 10 new AE classes, randomly selected from the K enrollment utterances. Note that to simplify the experiments, we performed adaptation simultaneously for 10 new AE classes. However, since we only update the embedding matrix, it is equivalent to performing adaptation for each new AE class at a time.

5.2. Experimental settings

For all the experiments, we adopted a Conv-TasNet-based network architecture, which consists of stacked dilated convolution blocks. We used the Asteroid toolkit for all experiments [25]. By following the notations of [26], we set the hyper-parameters as follows: N=256, L=20, B=256, H=512, P=3, X=8, and R=4. We also set the dimension of the embedding vectors

Table 1: *SDR improvement [dB] for experiment with mixtures of seen AE classes. The SDR of the mixture was -3.6 dB.*

Embedding at test time	**Model** 1-hot	Enrl	Mixed	Mixed+EL
1-hot	11.4	-	12.6	**12.9**
Enrl	-	10.4	**10.5**	10.1

Table 2: *SDR improvement [dB] for experiment on mixtures with new AE classes. The SDR for the mixture signals was -3.4 dB for the seen AEs and -4.0 dB for the new AEs. We used 1-hot-based encoder for extracting seen AEs except for the enrollment-based model, and Eq. (8) to compute the embedding vectors for the new AE classes.*

Model	**Seen AEs**	**New AEs** K=1	K=5	K=10
1-hot	10.5	-	-	-
Enrl	10.4	**4.9**	7.0	7.0
Mixed	11.6	4.4	7.2	7.5
Mixed + EL	**11.8**	3.2	7.1	7.4
+adapt (rnd init)	-	0.2	3.4	5.1
+adapt (avg init)	-	3.8	**7.8**	**8.2**

to D=256. The embedding vector is multiplied with the output of the first stacked convolution block (Figure 1). The enrollment encoder block consists of one stacked dilated convolution block (R=1) with the other hyper-parameters as for the extraction network. We used the Adam algorithm [27] for optimization with an initial learning rate of 10^{-4} and used gradient clipping [28]. For the experiments with EL, we set $\alpha = 3$. All models were trained for up to 200 epochs, and we used the models achieving the lowest cross-validation loss value in all experiments. During adaptation, we fixed all network parameters except for the new embedding vectors. We retrained the new embedding vectors for 10 epochs with a learning rate of 10^{-3}.

We evaluate the results in terms of scale-invariant signal-to-distortion ratio (SDR) computed with the BSSEval toolkit [29]. The results were obtained by averaging the SDR for each AE sound in each mixture of the test sets.

5.3. Results on mixtures of seen AE classes

Table 1 shows the SDR improvement for the test set consisting of seen AE classes. The table compares results with 1-hot ("1-hot") and enrollment ("Enrl")-based SoundBeam and the proposed SoundBeam-mixed, with and without EL, when using 1-hot- or enrollment-based embedding vectors at test time. 1-hot- and enrollment-based models serve as baselines for target sound extraction [7, 8].

The results confirm that the 1-hot based approach, which performs direct optimization of the embedding vectors, outperforms the enrollment-based approach. The proposed SoundBeam-mixed models further improve SDR by more than 1 dB. The model trained with EL achieves the best performance when using the 1-hot encoder. These results demonstrate the positive effect of the proposed multi-task training scheme on the seen classes.

5.4. Results on mixtures with new AE classes

Table 2 shows the SDR improvement for mixtures with new AE classes for the 1-hot, enrollment, and mixed models without and with adaptation. The left part of the table shows the extraction performance of the two seen AE classes in the mixtures. For the extraction of seen AE classes, we used 1-hot embedding vectors for all models except for the enrollment-based model. These results demonstrate that all models can extract sounds of the seen classes even if the mixtures include other sounds unseen during training.

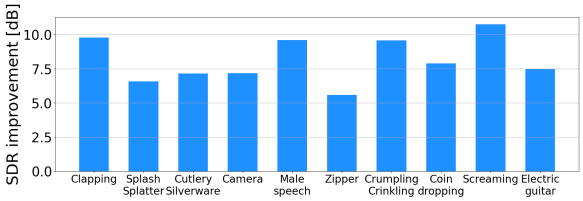

Figure 2: *SDR improvement for SoundBeam-mixed with EL and adaptation for the 10 new AE classes with K=10.*

The right-side of Table 2 shows the SDR improvement for the extraction of sounds from new AE classes for different numbers of enrollment samples. We used Eq. (8) to compute the embedding vectors for the new classes. The results demonstrate that the proposed mixed models also improve extraction performance for the new classes when $K > 1$. With a single enrollment sample (K=1), the sample may not well represent the new AE class causing a mismatch between the enrollment and the target sound. Consequently, the average extraction performance is relatively low. However, with more enrollment samples (K=5, 10) extraction performance greatly improves especially with SoundBeam-mixed, which achieves an SDR improvement of up to 7.5 dB.

Finally, the last rows of table 2 show the effect of adaptation ("+adapt") by retraining the embedding vectors associated with the new AE classes (see section 3). We considered two configurations for adaptation, with embedding randomly initialized ("rand init") or initialized with the averaged embedding vectors as shown in Eq. (8) ("avg init"). Adaptation from randomly initialized embedding vectors achieves an SDR improvement of up to about 5.1 dB, which is less than the 7.4 dB improvement obtained with the mixed model with averaged embedding vectors without adaptation. Adapting the averaged embedding vectors improves performance compared by up to 0.8 dB.

Figure 2 plots the SDR improvement for the new AE classes with the proposed SoundBeam-mixed with adaptation. The proposed scheme could achieve an SDR improvement of more than 5 dB on all 10 new classes. This result demonstrates that the proposed scheme can realize efficiently few-shot learning of target sound extraction for new AE classes.

6. Conclusion

We have proposed a novel target sound extraction system that combines both 1-hot- and enrollment-based approaches. We showed that by designing a sound extraction model with both 1-hot and enrollment encoders and a shared extraction network, we can enforce both approaches to share a common embedding space. This mixed model improves significantly extraction performance thanks to multi-task training. Moreover, it enables 1-hot based target sound extraction on new classes. Combined with retraining-based adaptation, we could achieve high extraction performance on new AE classes with only a few samples.

In future works, we will consider training the system on a larger dataset, which may include audio clips with multiple AE classes, or unlabeled data [8, 9]. Moreover, we will also investigate other EL [13] to provide more discriminative embedding vectors and extend the model to perform online processing [30].

7. References

[1] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Proc. of Interspeech*, 2017, pp. 2655–2659.

[2] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Cernocky, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[3] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," in *Proc. of Interspeech*, 2018, pp. 307–311.

[4] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *Proc. of Interspeech*, 2019, pp. 2728–2732.

[5] O. Slizovskaia, G. Haro, and E. Gómez, "Conditioned source separation for music instrument performances," *arXiv preprint arXiv:2004.03873*, 2021.

[6] J. H. Lee, H. Choi, and K. Lee, "Audio query-based music source separation," in *Proc. of ISMIR*, 2019, pp. 878–885.

[7] T. Ochiai, M. Delcroix, Y. Koizumi, H. Ito, K. Kinoshita, and S. Araki, "Listen to what you want: Neural network-based universal sound selector," in *Proc. of Interspeech*, 2020, pp. 1441–1445.

[8] B. Gfeller, D. Roblek, and M. Tagliasacchi, "One-shot conditional audio filtering of arbitrary sounds," in *Proc. of ICASSP*, 2021, pp. 501–505.

[9] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *Proc. of ICASSP*, 2020, pp. 101–105.

[10] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Proc. of DCASE*, 2018.

[11] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *Proc. of ISMIR*, 2017, pp. 486–493.

[12] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, "Improving speaker discrimination of target speech extraction with time-domain speakerbeam," in *Proc. of ICASSP*, 2020, pp. 691–695.

[13] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[14] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. of ICASSP*, 2019, pp. 626–630.

[15] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *Proc. of WASPAA*, 2019, pp. 175–179.

[16] E. Tzinis, S. Wisdom, A. Jansen, J. R. Hershey, A. J. Ellis, and P. Daniel, "Improving universal sound separation using sound classification presentation," in *Proc. of ICASSP*, 2020, pp. 96–100.

[17] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda, "Acoustic event detection method using semi-supervised non-negative matrix factorization with a mixture of local dictionaries," in *Proc. of DCASE*, 2016.

[18] K. M. Jeon, H. Kim, and H. K. Kim, "Nonnegative matrix factorization-based source separation with online noise learning for detection of rare sound events," in *Proc. of DCASE*, 2017.

[19] Y. Huang, L. Lin, S. Ma, X. Wang, H. Liu, Y. Qian, M. Liu, and K. Ouch, "Guided multi-branch learning systems for DCASE 2020 task 4," DCASE 2020 Challenge, Tech. Rep., 2020.

[20] http://dcase.community/challenge2020/task-sound-event-detection-and-separation-in-domestic-environments (accessed on April 2nd, 2021).

[21] F. Pishdadian, G. Wichern, and J. Le Roux, "Learning to separate sounds from weakly labeled scenes," in *Proc. of ICASSP*, 2020, pp. 91–95.

[22] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj *et al.*, "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 7, 2016.

[23] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. of WASPAA*, 2017, pp. 344–348.

[24] S. Wisdom, H. Erdogan, D. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. Hershey, "What's all the fuss about free universal sound separation data?" in *Proc. of ICASSP*, 2021, pp. 186–190.

[25] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martin-Doñas, D. Ditter, A. Frank, A. Deleforge, and E. Vincent, "Asteroid: The PyTorch-Based Audio Source Separation Toolkit for Researchers," in *Proc. of Interspeech*, 2020, pp. 2637–2641.

[26] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 8, pp. 1256–1266, 2019.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[28] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. of ICML*, 2013, pp. 1310–1318.

[29] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing (TASLP)*, vol. 14, no. 4, pp. 1462–1469, 2006.

[30] E. Tzinis, Z. Wang, X. Jiang, and P. Smaragdis, "Compute and memory efficient universal sound source separation," *arXiv preprint arXiv:2103.02644*, 2021.