IMPROVING SOUND EVENT DETECTION IN DOMESTIC ENVIRONMENTS USING SOUND SEPARATION

Nicolas Turpault[1]*, Scott Wisdom*[2]*, Hakan Erdogan*[2]*, John R. Hershey*[2]*,*
Romain Serizel[1]*, Eduardo Fonseca*[3]*, Prem Seetharaman*[4]*, Justin Salamon*[5]

[1] Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
[2] Google Research, AI Perception, Cambridge, United States
[3] Music Technology Group, Universitat Pompeu Fabra, Barcelona
[4] Interactive Audio Lab, Northwestern University, Evanston, United States
[5] Adobe Research, San Francisco, United States

ABSTRACT

Performing sound event detection on real-world recordings often implies dealing with overlapping target sound events and non-target sounds, also referred to as interference or noise. Until now these problems were mainly tackled at the classifier level. We propose to use sound separation as a pre-processing for sound event detection. In this paper we start from a sound separation model trained on the Free Universal Sound Separation dataset and the DCASE 2020 task 4 sound event detection baseline. We explore different methods to combine separated sound sources and the original mixture within the sound event detection. Furthermore, we investigate the impact of adapting the sound separation model to the sound event detection data on both the sound separation and the sound event detection.

Index Terms— Sound event detection, synthetic soundscapes, sound separation

1. INTRODUCTION

Sound event detection (SED) is the task of describing, from an audio recording, what happens and when each single sound event is occurring [1]. This is something that we, as humans, do rather naturally to obtain information about what is happening around us. However, trying to reproduce this with a machine is not trivial, as the SED algorithm needs to cope with several problems, including audio signal degradation due to additive noise or overlapping events [2]. Indeed, in real-world scenarios, the recordings provided to the SED systems contain not only target sound events, but also sound events that can be considered as "noise" or "interference." Also, several target sound events can occur simultaneously.

In the past, the overlapping sound events problem has been tackled from the classifier point of view. This can be done by training the SED as a multilabel system in which case the most energetic sound events are usually detected more accurately than the rest [3, 4]. Some other approaches tried to deal more explicitly with this problem using either a set of binary classifiers [5], using factorization techniques on the input of the classifier [6, 7], or exploiting spatial information when available [8]. The additive noise problem is usually solved by training SED systems on noisy signals. This may be effective to some degree when the noise level is low, but much less so when the noise level increases [4].

Sound separation (SSep) seems like a natural candidate to solve these two issues [9, 10]. SSep systems are trained to predict the constituent sources directly from mixtures. Thus, sound separation can both decrease the level of interfering noise and enable a SED system to detect quieter events in overlapping acoustic mixtures. Until recently, SSep has been mainly applied to specific classes of signals, such as speech or music. However, recent works has shown that sound separation can also be applied to separating sounds of arbitrary classes, a task known as "universal sound separation" [11, 12, 13].

In this paper, we propose to combine a universal SSep algorithm [11, 12] used as a pre-processing to the DCASE 2020 SED baseline [14]. We investigate the impact of the data used to train the SSep on the SED performance. We also explore different ways to combine the separated sound sources at different stages of SED.

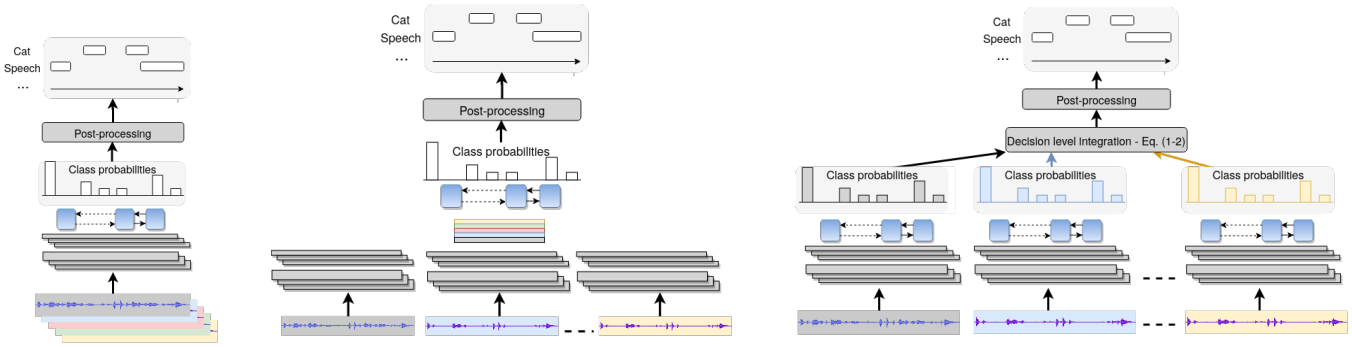
2. PROBLEM AND BASELINES DESCRIPTION

We aim to solve a problem similar to that of DCASE 2019 Task 4 [15]. Systems are expected to produce strongly-labeled outputs (i.e. detect sound events with a start time, end time, and sound class label), but are provided with weakly labeled data (i.e. sound recordings with only the presence/absence of a sound event included in the labels without any timing information) for training. Multiple events can be present in each audio recording, including overlapping target sound events and potentially non-target sound events. Previous studies have shown that the presence of additional sound events can drastically decrease the SED performance [4].

2.1. Sound event detection baseline

The SED baseline system uses a mean-teacher model which is a combination of two models: a student model and a teacher model (both have the same architecture). The student model is the final model used at inference time, while the teacher model is aimed at helping the student model during training and its weights are an exponential moving average of the student model's weights. A more detailed description can be found in Turpault and Serizel [14].

Part of this work was made with the support of the French National Research Agency, in the framework of the project LEAUDS "Learning to under-stand audio scenes" (ANR-18-CE23-0020) and the French region Grand-Est. Experiments presented in this paper were carried out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).



(a) Early integration. (b) Middle integration. (c) Late integration.

Figure 1: Integration between the SSep and SED (gray waveform represents mixture, and colored waveforms represent separated sources).

2.2. Sound separation baseline

The baseline SSep model uses a similar architecture to an existing approach for universal sound separation with a fixed number of sources [11, 12], which employs a convolutional masking network using STFT and analysis and synthesis. The training loss is negative stabilized signal-to-noise ratio (SNR) [16] with a soft-threshold SNR_{max}. Going beyond previous work, the model in this paper is able to handle variable number sources by using different loss functions for active and inactive reference sources that encourage the model to only output as many nonzero sources as exist in the mixture. Additional source slots are encouraged to be all-zero.

3. SOUND EVENT DETECTION AND SEPARATION

3.1. Sound separation for sound event detection

Overlapping sound events are typically more difficult to detect as compared to isolated ones. SS can be used for SED by first separating the component sounds in a mixed signal and then applying SED on each of the separated tracks [9, 10]. The decisions obtained on separated signals may be more accurate than the ones on the mixed signal. On the other hand, separation of sounds is not a trivial problem and may introduce artifacts which in turn may make sound SED harder. So, it is necessary to jointly investigate SSep and SED.

3.2. Sound event detection on separated sources

In the approaches described here, SSep provides several audio clips that contain information related to the sound sources composing the original (mixture) clip. Each of these new audio clips (separated sound sources) are used together with the mixture clip within the SED. We compare three different approaches to integrate the information from these audio clips at different levels of the model.

3.2.1. Early integration

This approach is similar to the SED baseline except that all the audio clips (mixture and separated sound sources) are concatenated as input channels to form a new tensor (Figure 1a). The first channel always contains the mixture clip while the separated sound source clips are provided with no particular order. The model is trained like the SED baseline using the annotations of the mixture clip.

3.2.2. Middle integration

We re-use the CNN block from the SED baseline to extract embeddings from the mixture clip and the separated sound sources clips (Figure 1b). The embeddings are concatenated along the feature axis and fed into a fully connected layer before training a new RNN classifier within a mean-teacher student framework.

3.2.3. Late integration

For this approach, we apply the SED baseline on the mixture clip and the separated sound sources (Fig. 1c). The SED output for each of these clips are obtained from the $y_{\text{DSL},c}$, the raw outputs of the classifier corresponding to each sound class c among the C sound classes. The combined raw output (before thresholding and post-processing) for each class c is obtained as follows:

$$y_{\text{DSL},c} = \Big(\frac{y_{M,c}^{q} + y_{SS,c}^{q}}{2}\Big)^{1/q} \tag{1}$$

where $y_{M,c}$ and $y_{SS,c}$ are the raw classifier outputs for the sound class c obtained on the mixture clips and the separated sound sources, respectively. The sound source/mixture combination weight is q. The classifier output for the sound class c is obtained from the raw classifier outputs on each individual separated sound sources as follows:

$$y_{SS,c} = \Big(\frac{1}{N_s}\sum_{s=1}^{N_s} \mathbf{y}_{s,c}^{p}\Big)^{1/p} \tag{2}$$

where N_s is the number of separated sound source clips obtained from the SSep, $y_{s,c}$ is the raw classifier output for the sound class c obtained for the separated sound source clip s and p is the sound sources combination weight.

4. BASELINES SETUP AND DATASET

4.1. DESED dataset

The dataset used for the SED experiments is DESED[1], a dataset for SED in domestic environments composed of 10-sec audio clips that are recorded or synthesized [15, 4]. The recorded soundscapes are taken from AudioSet [17]. The synthetic soundscapes are generated using Scaper [18]. The foreground events are obtained from

[1] https://project.inria.fr/desed/

Table 1: Performance for the SED baseline [14] on DESED.

	F1-Score	PSDS
REC_VAL	37.8	0.540
REC_EVAL	39.0	0.552
SYN_VAL	62.6	0.695

FSD50k [19, 20]. The background textures are obtained from the SINS dataset [21] and TUT scenes 2016 development dataset [22].

The dataset includes a synthetic validation set simulated from different isolated those in the training set (SYN_VAL) , a validation set and a public evaluation set composed of recorded clips (REC_VAL and REC_EVAL) that are used to adjust the hyper-parameters and evaluate the SED, respectively.

4.2. FUSS dataset

The Free Universal Sound Separation (FUSS)[2] dataset [23] is intended for experimenting with universal sound separation [11], and is used as training data for the SSep system. Audio data is sourced from `freesound.org`. Using labels from FSD50k [20], gathered through the Freesound Annotator [24], these source files have been screened such that they likely only contain a single type of sound. Labels are not provided for these source files, and thus the goal is to separate sources without using class information. To create reverberant mixtures, 10 second clips of sources are convolved with simulated room impulse responses. Each 10 second mixture contains between 1 and 4 sources. Source files longer than 10 seconds are considered "background" sources. Every mixture contains one background source, which is active for the entire duration.

4.3. Sound event detection baseline

The SED baseline[3] architecture and parameters are described extensively in Turpault et al. [14]. The performance obtained with this baseline on DESED is presented in Table 1.

4.4. Sound separation baseline

The SSep system is trained on 16-kHz audio[4]. The input to the SSep network is the magnitude of the STFT using window size 32ms and hop of 8ms. These magnitudes are processed by an improved time-domain convolutional network (TDCN++) [11, 12], which is similar to Conv-TasNet [25]. Like Conv-TasNet, the TDCN++ consists of four repeats of 8 residual dilated convolution blocks, where within each repeat the dilation of block ℓ is 2^ℓ for $\ell = 0, .., 7$. The main differences between the TDCN++ and Conv-Tasnet are (1) bin-wise normalization instead of global layer normalization, which averages only over basis frames instead of frames and frequency bins, (2) trainable scalar scale parameters multiplied after each dense layer, which are initialized with 0.9^i, and (3) additional residual connections between blocks, with connection pattern $0 \rightarrow 8$, $0 \rightarrow 16$, $0 \rightarrow 24$, $8 \rightarrow 16$, $8 \rightarrow 24$, $16 \rightarrow 24$.

[2] `https://github.com/google-research/sound-separation/tree/master/datasets/fuss`

[3] `https://github.com/turpaultn/dcase20_task4/tree/papers_code`

[4] `https://github.com/google-research/sound-separation/tree/master/models/dcase2020_fuss_baseline`

Table 2: SSep and SED performance for FUSS-trained SSep models: MSi (multi-source SI-SNR improvement) and 1S (single-source SI-SNR). Confidence intervals: ± 1.2 (F1), ± 0.015 (PSDS).

	FUSS test set				REC_VAL	
FUSS	Rev.		Dry		Late Integration	
training	MSi	1S	MSi	1S	F1-Score	PSDS
Rev.	**12.5**	**37.6**	**10.4**	**32.1**	38.2	0.565
Dry	10.4	31.2	10.2	31.8	**39.2**	**0.574**

Table 3: DESED+FUSS tasks.

Task	Sources
DmFm	DESED mix, dry FUSS mix
BgFgFm	DESED bg, DESED fg mix, dry FUSS mix
PIT	DESED bg, dry FUSS mix, 5 DESED fg sources
Classwise	DESED bg, 10 DESED classes, dry FUSS mix
GroupPIT	DESED bg, 5 DESED fg sources, 4 dry FUSS srcs

This TDCN++ network predicts four masks that are the same shape as the input STFT. Each mask is multiplied with the complex input STFT, and a source waveform is computed by applying the inverse STFT. A weighted mixture consistency projection layer [26] is applied to the separated waveforms to be consistent with the input mixture waveform where the per-source weights are predicted by an additional dense layer using the penultimate output of TDCN++.

To separate mixtures with variable numbers of sources, different loss functions are used for active and inactive reference sources. For active reference sources (i.e. non-zero reference source signals), the soft-threshold for SNR is 30 dB, equivalent to the error power being below the reference power by 30 dB. For non-active reference sources (i.e. all-zero reference source signals), the soft-threshold is 20 dB measured relative to the mixture power, thus gradients are clipped when the error power is 20 dB below the mixture power. Thus, for a N-source mixture, a M-output model with $M \geq N$ should output M non-zero sources, and $M - N$ all-zero sources.

4.5. Evaluation metrics

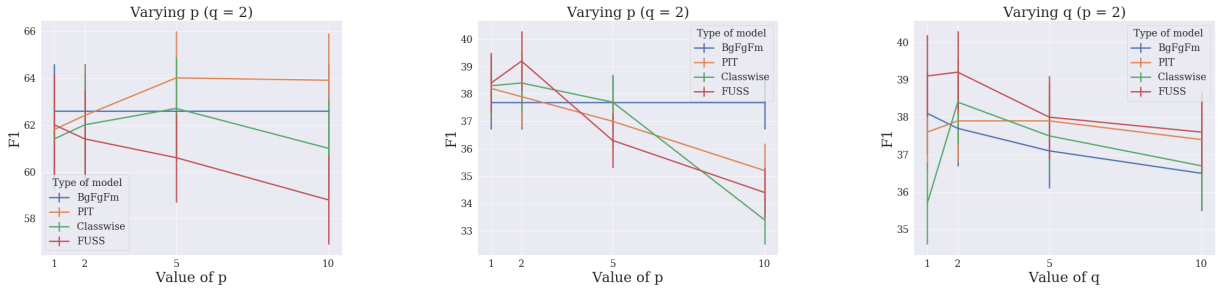
SSep systems are evaluated in terms of scale-invariant SNR (SI-SNR) [27]. Since FUSS mixtures can contain one to four sources, we report two scores to summarize performance: multi-source SI-SNR improvement (MSi), which measures the separation quality of mixtures with two or more sources, and single-source SI-SNR (1S), which measures the separation model's ability to reconstruct single-source inputs.

SED systems are evaluated according to an event-based F1-score with a 200 ms collar on the onsets and a collar on the offsets that is the greater of 200 ms and 20% of the sound event's length. The overall F1-score is the unweighted average of the class-wise F1-scores (macro-average). F-scores are computed on a single operating point (decision thresholds=0.5) using the sed_eval library [28].

SED systems are also evaluated with polyphonic sound event detection scores (PSDS) [29]. PSDS are computed using 50 operating points (linearly distributed from 0.01 to 0.99) with the following parameters: detection tolerance parameter ($\rho_{\mathrm{DTC}} = 0.5$), ground truth intersection parameter ($\rho_{\mathrm{GTC}} = 0.5$), cross-trigger tolerance parameter ($\rho_{\mathrm{CTTC}} = 0.3$), and maximum false positive rate ($e_{\max} = 100$). The weight on the cross-trigger and class instability costs are set to $q_{\mathrm{CT}} = 1$ and $q_{\mathrm{ST}} = 0$, respectively.

Table 4: SSep and SED performance for various SSep tasks. "Bg" is DESED background, "Fg" is DESED foreground, and "Fm" is FUSS mixture. Confidence intervals: ± 1.2 (F1-score) and ± 0.015 (PSDS) on the validation set and ± 2.3 (F1-score) on the synthetic set.

SS training task	BgFgFm validation SI-SNRi (dB)			REC_VAL						SYN_VAL
				Early integration		Middle integration		Late integration		Late integration
	Bg	Fg	Fm	F1-score	PSDS	F1-Score	PSDS	F1-Score	PSDS	F1-score
DmFm	1.8	0.1	17.3	**35.4**	**0.545**	**35.9**	**0.548**	36.2	0.573	62.3
BgFgFm	**18.3**	**18.4**	**17.5**	35.1	0.529	33.2	0.531	37.7	0.568	**62.6**
PIT	17.2	17.6	17.3	31.6	0.461	33.2	0.472	37.9	**0.574**	62.4
Classwise	16.8	17.5	**17.5**	27.4	0.361	28.9	0.386	**38.4**	0.566	62.0
GroupPIT	16.8	18.0	17.2	31.6	0.486	32.3	0.473	38.2	0.570	62.2



(a) Impact of p (SYN_VAL) (b) Impact of p (REC_VAL) (c) Impact of q (REC_VAL)

Figure 2: impact of the late integration weights on the SED performance (vertical bars represent confidence intervals)

5. EXPERIMENTS

Table 2 displays SSep and SED performance on the FUSS test set and REC_VAL. For SSep we do a full cross-evaluation between the dry and reverberant versions. From this we can see that the reverberant FUSS-trained model achieves the best separation scores across both dry and reverberant conditions. However, in terms of SED performance, the dry FUSS-trained separation model yields the best performance in terms of both F1 and PSDS. This may be due to the synthetic room impulse responses used to create reverberant FUSS being mismatched to the real data in REC_VAL. Thus, we opt to use the dry version of FUSS in the proceeding experiments.

Besides training SSep systems on FUSS, we also constructed a number of tasks consisting of data from both DESED and FUSS, described in Table 3. Some tasks are trained with permutation-invariant training (PIT) [30] or groupwise PIT.

Table 4 reports the results of evaluating these models on the BgFgFm task. For models with more than three outputs, the sources for corresponding classes are summed together. For example, sources 1 through 5 are summed together for the PIT and GroupPIT models, and sources 0 through 9 for the Classwise model, to produce the separated estimate of the DESED foreground mixture. The BgFgFm-trained SSep model achieves the best SSep scores, since it is matched to the task. This model also achieves the highest F1 score on the SYN_VAL set, although this is not statistically significant. However, on REC_VAL, the Classwise model achieves the best F1 score. However, notice that the dry FUSS SSep model achieves the overall best F1 and PSDS scores of 39.2 and 0.574 in Table 2. This suggests that the DESED+FUSS-trained SSep models do not generalize as well, since they are trained on more specific synthetic data compared to FUSS-trained models.

Figure 2 displays the impact of the late integration parameters p and q on the SED performance. Intuitively when the SSep models aims at separating sources that corresponds to target sound events, the parameter p should be high so the source aggregation is close to a max pooling across sources. This is what can be observed on Figure 2a for the PIT model. For the FUSS-trained SSep separated sources do not correspond to target sources and the integration is better for low values of p. This is not confirmed on REC_VAL (Fig. 2b). This could be due to the mismatch between training and test for the SSep leading to sources that are not properly separated.

The SED performance depending on the parameter q is presented on Figure 2c. A high value for the parameter q means focusing only on the mixture or on the separated sounds and leads to degraded performance for all the SSep models. The best performance is then obtained with the FUSS-trained SSep and $p = 2$ and $q = 2$ (40.7% F1-score and 0.570 PSDS on REC_EVAL).

6. CONCLUSION

In this paper we proposed to use a SSep algorithm as pre-processing to a SED system applied to complex mixtures including non-target events and background noise. We proposed to retrain the generic SSep on task specific datasets. The combination has shown to have potential to improve the SED performance in particular when using a late integration to combine the prediction obtained from the separated sources. However, the benefits still remain limited most probably because of the mismatch between the SSep training conditions and the SED test conditions.

7. ACKNOWLEDGEMENTS

We would like to thank the other organizers of DCASE 2020 task 4: Daniel P. W. Ellis and Ankit Parag Shah.

8. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[2] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *ICASSP*.

[3] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 724–728.

[4] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. ICASSP*, 2020.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[6] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6450–6454.

[7] V. Bisot, S. Essid, and G. Richard, "Overlapping sound event detection with supervised nonnegative matrix factorization," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 31–35.

[8] S. Adavanne, A. Politis, and T. Virtanen, "Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[9] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8677–8681.

[10] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 101–105.

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

[12] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. Ellis, "Improving universal sound separation using sound classification," in *Proc. ICASSP*, 2020.

[13] M. Olvera, E. Vincent, R. Serizel, and G. Gasso, "Foreground-Background Ambient Sound Scene Separation," May 2020, working paper or preprint.

[14] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," 2020, working paper or preprint.

[15] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. DCASE Workshop*, 2019.

[16] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey, "Unsupervised sound separation using mixtures of mixtures," *arXiv preprint arXiv:2006.12701*, 2020.

[17] 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. ICASSP*, 2017.

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

[19] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. ACM*, 2013, pp. 411–412.

[20] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50k: an open dataset of human-labeled sound events," in *arXiv*, 2020.

[21] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. DCASE Workshop*, November 2017, pp. 32–36.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1128–1132.

[23] S. Wisdom, H. Erdogan, D. P. W. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. R. Hershey, "What's all the FUSS about free universal sound separation data?" *In preparation*, 2020.

[24] 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. ISMIR)*, 2017, pp. 486–493.

[25] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," vol. 27, no. 8, pp. 1256–1266, 2019.

[26] S. Wisdom, J. R. Hershey, K. Wilson, J. Thorpe, M. Chinen, B. Patton, and R. A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. ICASSP*, 2019.

[27] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 626–630.

[28] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, May 2016.

[29] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *Proc. ICASSP*, 2020.

[30] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017, pp. 241–245.