What's All the FUSS About Free Universal Sound Separation Data?

Scott Wisdom, Hakan Erdogan, Daniel Ellis, Romain Serizel, Nicolas Turpault, Eduardo Fonseca, Justin Salamon, Prem Seetharaman, John Hershey

WHAT'S ALL THE FUSS ABOUT FREE UNIVERSAL SOUND SEPARATION DATA?

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

[1]Google Research, United States
[2]Université de Lorraine, CNRS, Inria, Loria, Nancy, France
[3]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[4]Adobe Research, San Francisco, United States
[5]Descript, Inc., San Francisco, United States

ABSTRACT

We introduce the *Free Universal Sound Separation* (FUSS) dataset, a new corpus for experiments in separating mixtures of an unknown number of sounds from an open domain of sound types. The dataset consists of 23 hours of single-source audio data drawn from 357 classes, which are used to create mixtures of one to four sources. To simulate reverberation, an acoustic room simulator is used to generate impulse responses of box shaped rooms with frequency-dependent reflective walls. Additional open-source data augmentation tools are also provided to produce new mixtures with different combinations of sources and room simulations. Finally, we introduce an open-source baseline separation model, based on an improved time-domain convolutional network (TDCN++), that can separate a variable number of sources in a mixture. This model achieves 9.8 dB of scale-invariant signal-to-noise ratio improvement (SI-SNRi) on mixtures with two to four sources, while reconstructing single-source inputs with 35.5 dB absolute SI-SNR. We hope this dataset will lower the barrier to new research and allow for fast iteration and application of novel techniques from other machine learning domains to the sound separation challenge.

***Index Terms*—** Universal sound separation, variable source separation, open-source datasets, deep learning

1. INTRODUCTION

Hearing is often confounded by the problem of interference: multiple sounds can overlap and obscure each other, making it difficult to selectively attend to each sound. In recent years this problem has been addressed by using deep neural networks to extract sounds of interest, separating them from a mixture. Sound separation has made significant progress by focusing on constrained tasks, such as separating speech versus non-speech, or separating one speaker from another in a mixture of speakers, often with assumed prior knowledge of the number of sources to be separated. However human hearing seems to require no such limitations, and recent work has engaged in *universal sound separation*: the task of separating mixtures into their component sounds, regardless of the number and types of sounds.

One major hurdle to training models in this domain is that even if high-quality recordings of sound mixtures are available, they cannot be easily annotated with ground truth. High-quality simulation is one approach to overcome this limitation. Achieving good results requires supervised training using ground-truth source signals, drawn from a diverse set of sounds, and mixed using a realistic room simulator. Although previous efforts have created open-domain data [1], the number of sources was still considered known, and the impact on the field was limited by proprietary licensing requirements.

To make such data widely available, we introduce the *Free Universal Sound Separation* (FUSS) dataset. FUSS relies on CC0-licensed audio clips from `freesound.org`. We developed our own room simulator that generates the impulse response of a box shaped room with frequency-dependent reflective properties given a sound source location and a microphone location. As part of the dataset release, we also provide pre-calculated room impulse responses used for each audio sample along with mixing code, so the research community can simulate novel audio without running the computationally expensive room simulator.

Finally, we have released a masking-based baseline separation model, based on an improved time-domain convolutional network (TDCN++), described in our recent publications [1, 2]. On the FUSS test set, this model achieves 9.8 dB of scale-invariant signal-to-noise ratio improvement (SI-SNRi) on mixtures with two to four sources, while reconstructing single-source inputs with 35.5 dB SI-SNR.

Source audio, reverb impulse responses, reverberated mixtures and sources, and a baseline model checkpoint are available for download[1]. Code for reverberating and mixing audio data and for training the released model is available on GitHub[2].

The dataset was used in the DCASE 2020 challenge, as a component of the Sound Event Detection and Separation task. The released model served as a baseline for this competition, and a benchmark to demonstrate progress against in future experiments.

We hope this dataset will lower the barrier to new research, and particularly will allow for fast iteration and application of novel techniques from other machine learning domains to the sound separation challenge. This paper provides three main contributions:

1. We describe a new free and open-source dataset for universal sound separation, which is the largest to date in terms of both amount of data (23 hours of single-source audio data) and number of classes (357), and includes a variety of conditions, including variable source number (1-4) and reverberation.
2. We propose new loss functions and evaluation metrics for models that can separate variable numbers of sources.
3. We provide a baseline implementation of an open-domain separation system designed for variable numbers of sources, to serve as a benchmark for future work.

[1]`https://zenodo.org/record/4012661`
[2]`https://git.io/JTusI`

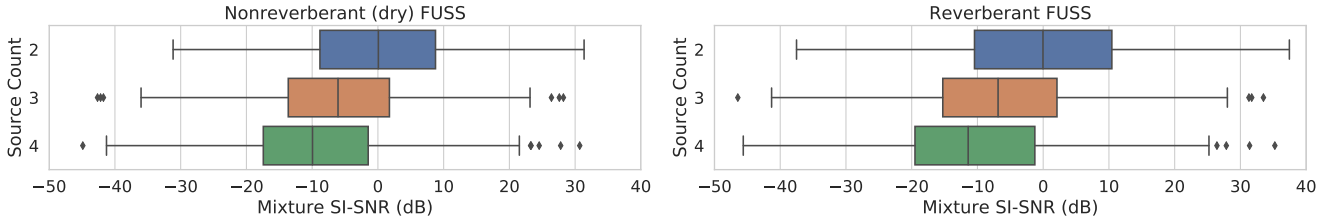

Fig. 1. Box plots for input SI-SNR for each source count on dry (left) and reverberant (right) FUSS.

2. RELATION TO PRIOR WORK

Only recently was open-domain (i.e. hundreds of sound classes) universal sound separation shown to be feasible [1], though only with a fixed number of sources, and on proprietary data with restrictive licensing. Zadeh et al. [3] constructed a small (less than 1 hour) dataset with 25 sound classes and proposed a transformer-based model to separate a fixed number of sources. Tzinis et al. [4] performed separation experiments with a fixed number of sources on the 50-class ESC-50 dataset [5]. Other papers have leveraged information about sound class, either as conditioning information or as as a weak supervised signal [6, 2, 7].

In terms of variable source separation, Kinoshita et al. [8] proposed another approach for handling variable numbers of sources, where the separation network is made recurrent over sources. One drawback of this approach compared to ours is that the source-recurrent network needs to be run N times to separate N sources, while our proposed network only need to run once. More recently, Luo and Mesgarani [9] proposed a separation model trained to predict the mixture for inactive output sources. In contrast, our approach trains the separation model to predict zero for inactive sources, making it easier to determine when sources are inactive.

One typical application of universal sound separation is sound event detection (SED). In real scenarios SED systems commonly have to deal with complex soundscapes with possibly overlapping target sound events and non-target interfering sound events. Approaches have been developed to deal with overlapping target sound events using multilabeled (foreground vs background) training sets [10], sets of binary classifiers [11], factorization techniques on the input of the classifier [12, 13], or exploiting spatial information when available [14]. However, none of these approaches explicitly solved the problem of non-target events. Sound separation 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 [15, 7, 16, 17, 18].

3. DATA PREPARATION

The audio data is sourced from a subset of FSD50K [19], a sound event dataset composed of `freesound.org` content annotated with labels from the AudioSet ontology [20]. Audio clips are of variable length ranging from 0.3 to 30s, and labels were gathered through the Freesound Annotator [21]. Sound source files were selected which contain only one class label, so that they likely only contain a single type of sound. After also filtering for Creative Commons public domain (CC0) licenses, we obtained about 23 hours of audio, consisting of 12,377 source clips useful for mixing.

To create mixtures, these sounds were split by uploader and further partitioned into 7,237 for training, 2,883 for validation, and 2,257 for evaluation. Using this partition we created 20,000 training mixtures, 1,000 validation mixtures, and 1,000 evaluation mixtures, each 10s in length. The sampling procedure is as follows. For each mixture, the number of sounds is chosen uniformly at random in the range of one to four. Every mixture contains zero to three *foreground* source events, and one *background* source, which is active for the entire duration. The foreground and background sounds are sampled, using rejection sampling, such that each source in a mixture has a different sound class label. For foreground sounds, a sound source file less than 10s long is sampled, and the entire sound is placed uniformly at random within the 10s clip. For the background source, a file longer than 10s is sampled uniformly with replacement, and a 10s segment is chosen uniformly at random from within the file. Choosing one longer source as a background for each mixture does introduce some bias in the class distribution of examples (examples of the biased classes include natural sounds such as wind, rain, and fire and man-made sounds such as piano, engine, bass guitar, and acoustic guitar), but it has the benefit of avoiding an unrealistic pattern of pure silence regions in every example. Note that the background files were not screened for silence regions, and hence some of the background sounds may still contain significant periods of silence. Figure 1 shows the distribution of input SI-SNR for examples with 2, 3, and 4 sources. Table 1 shows the proportion of local overlap, as measured with non-overlapping 25 ms windows. From this table it is clear that the background sources are active most of the time 81%), and mixtures with two or more sources contain sounds that do not always completely overlap.

Table 1. Local overlap amount ($\%$) per source count.

	Dry FUSS					**Rev FUSS**				
Count	0	1	2	3	4	0	1	2	3	4
1	19	81				23	77			
2	13	63	24			20	59	21		
3	8	47	36	9		11	45	35	8	
4	7	36	34	20	4	10	35	33	19	3

The mixing procedure is implemented using Scaper [22][3]. Scaper is a flexible tool for creating source separation datasets – sound event parameters such as SNR, start time, duration, data augmentation (e.g., pitch shifting and time stretching), among others, can be sampled from user-defined distributions, allowing us to programmatically create and augment randomized, annotated soundscapes. This allows FUSS to easily be extended by mixing larger amounts of training data and adding new source data and mixing conditions [23].

The room simulation is based on the image method with frequency-dependent wall filters and is described in [24]. A simulated room with width between 3-7 meters, length between 4-8 meters, and height between 2.13-3.05 meters is sampled for each mixture, with a random microphone location, and the sources in the clip are each convolved with an impulse response from a different randomly sampled location within the simulated room. The locations are sampled uniformly, with a minimum distance of 20

[3] We used Scaper 1.6.0 to generate FUSS.

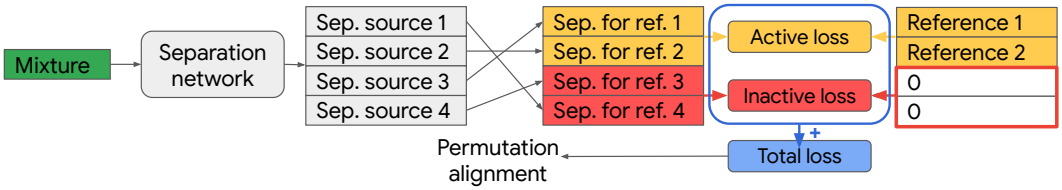

Fig. 2. Variable source separation for a separation model with $M = 4$ outputs and input mixture with $M_a = 2$ active references.

cm between each source and all the microphones. During impulse response generation, all source image locations are jittered by up to 8 cm in each direction to avoid the sweeping echo problem [25]. The wall materials of each room is chosen randomly from common materials with frequency-dependent acoustic reflectivities, where we also randomly use a gain factor between 0.5 and 0.95 to increase variability of RIRs and better match real room impulse responses. We generate 20,000 train, 1000 validation, and 1000 test rooms.

4. BASELINE MODEL

The baseline model uses a time-domain convolutional network (TDCN++) [1, 2, 26, 27] that incorporates several improvements over the Conv-TasNet model [28]. These improvements include feature-wise layer normalization over frames instead of global normalization, longer-range skip-residual connections, and a learnable scaling parameter after each dense layer initialized to an exponentially decaying scalar equal to 0.9^{ℓ}, where ℓ is the layer or block index. Using a short-time Fourier transform (STFT) with 32 ms window and 8 ms hop, input audio is transformed to a complex spectrogram. The magnitude of this spectrogram is fed to the TDCN++ network, which produces M masks via a fully-connected output layer. These masks are multiplied with complex spectrogram input, and initial separated sources $\underline{\mathbf{s}}$ are produced by applying an inverse STFT. Finally, a mixture consistency layer [29] is applied to these initial separated source waveforms:

$$\hat{s}_m = \underline{s}_m + \frac{1}{M}(x - \sum_{m'} \underline{s}_{m'}), \tag{1}$$

which projects the separated sources such that they sum up to the original input mixture x. Since we know that FUSS mixtures contain up to 4 sources, we choose to use $M = 4$.

4.1. Variable Source Separation

This baseline model is able to separate mixtures with a variable numbers of sources. To accomplish this, we propose a new loss function, as illustrated in Figure 2. Assume we have a training mixture x with M_a reference sources, where M_a can be less than the number of output sources M. Thus, the training mixture has $M_a \leq M$ active reference sources $\mathbf{s} \in \mathbb{R}^{M_a \times T}$, while the separation model produces separated sources $\hat{\mathbf{s}} \in \mathbb{R}^{M \times T}$. For such an example, we use the following permutation-invariant training loss:

$$\mathcal{L}(\mathbf{s}, \hat{\mathbf{s}}) = \min_{\pi \in \Pi} \Big[\sum_{m_a=1}^{M_a} \mathcal{L}_{\text{SNR}}(s_{m_a}, \hat{s}_{\pi(m_a)}) + \sum_{m_0=M_a+1}^{M} \mathcal{L}_0(x, \hat{s}_{\pi(m_0)}) \Big], \tag{2}$$

where the active per-source loss is negative SNR with a threshold $\tau = 10^{-\text{SNR}_{\max}/10}$ that determines the maximum allowed SNR [26]:

$$\mathcal{L}_{\text{SNR}}(y, \hat{y}) = 10 \log_{10} \left(\|y - \hat{y}\|^2 + \tau \|y\|^2 \right) \tag{3}$$

and the loss $\mathcal{L}_0$ for "inactive" separated sources is

$$\mathcal{L}_0(x, \hat{y}) = 10 \log_{10} \left(\|\hat{y}\|^2 + \tau \|x\|^2 \right). \tag{4}$$

Note that equation (4) is equivalent to $\mathcal{L}_{\text{SNR}}$ (3), with the reference y set to 0 and the thresholding performed using the mixture x instead of the reference source y.

Thus, separated sources are permuted to match up with either active reference sources $\mathbf{s}_a$ or all-zeros "references". If a separated source is paired with an all-zeros reference, the $\mathcal{L}_0$ loss function (4) is used, which minimizes the source's power until the ratio of mixture power to separated source power drops below a threshold. When the inactive loss $\mathcal{L}_0$ is small, this threshold prevents the large inactive loss gradients from dominating the overall loss. We found that 30 dB was a good value for $\text{SNR}_{\max}$ in order to set τ.

5. EVALUATION METRICS

To evaluate performance, we use scale-invariant signal-to-noise ratio (SI-SNR) [30]. SI-SNR measures the fidelity between a target y and an estimate $\hat{y}$ within an arbitrary scale factor in units of decibels:

$$\begin{aligned} \text{SI-SNR}(y, \hat{y}) &= 10 \log_{10} \frac{\|\alpha y\|^2}{\|\alpha y - \hat{y}\|^2} \\ &\approx 10 \log_{10} \frac{\|\alpha y\|^2 + \epsilon}{\|\alpha y - \hat{y}\|^2 + \epsilon} \end{aligned} \tag{5}$$

with $\alpha = \text{argmin}_a \|ay - \hat{y}\|^2 = \frac{y^T \hat{y}}{\|y\|^2} \approx \frac{y^T \hat{y}}{\|y\|^2 + \epsilon}$, where a small positive stabilizer, ϵ, is used to avoid singularity. We found that this implementation of SI-SNR can lead to inaccuracies in the context of FUSS, especially in the case of *under-separation*, where there are fewer non-zero source estimates than there are references. In this case it produces optimistic scores when $\hat{y}$ is close to 0 due to imprecision in the estimated scale α. Our initial reported results on FUSS [31] used this overly optimistic measure. To correct for this, we use an alternate formulation,

$$\begin{aligned} \text{SI-SNR}(y, \hat{y}) &= 10 \log_{10} \frac{\rho^2(y, \hat{y})}{1 - \rho^2(y, \hat{y})} \\ &\approx 10 \log_{10} \frac{\rho^2(y, \hat{y}) + \epsilon}{1 - \rho^2(y, \hat{y}) + \epsilon}, \end{aligned} \tag{6}$$

where $\rho(y, \hat{y}) = \frac{y^T \hat{y}}{\|y\| \|\hat{y}\|} \approx \frac{y^T \hat{y}}{\|y\| \|\hat{y}\| + \epsilon}$ is the cosine similarity between y and $\hat{y}$, with stabilizer ϵ. This is equivalent to (5) when $\epsilon = 0$, but is more accurate when $\epsilon > 0$. In our experiments we use $\epsilon = 10^{-8}$.

To account for inactive (i.e. zero or near-zero) references and estimates, we use the following procedure during evaluation. Separated sources are aligned to reference signals by maximizing SI-SNR

Table 2. Separation results for baseline FUSS model in terms of single-source SI-SNR (1S) and multi-source SI-SNRi (MSi) for various source counts, and source-counting accuracy for under, equal, and over-separation.

			1S	MSi vs source count				Source count rate		
Eval	**Split**	**Train**	**1**	**2**	**3**	**4**	**2-4**	**Under**	**Equal**	**Over**
Dry	Val	Dry	34.2	9.9	11.1	8.8	9.8	0.23	0.61	0.16
		Rev	34.4	9.4	8.8	8.2	8.7	0.32	0.51	0.17
	Test	Dry	35.5	11.2	11.6	7.4	9.8	0.25	0.60	0.15
		Rev	38.4	10.9	9.0	7.7	9.0	0.32	0.54	0.15
Rev	Val	Dry	35.1	10.2	11.8	10.2	10.7	0.30	0.59	0.12
		Rev	58.4	11.5	12.1	11.9	11.9	0.39	0.51	0.11
	Test	Dry	36.9	10.7	11.9	7.7	9.9	0.29	0.60	0.12
		Rev	65.8	12.7	11.6	10.2	11.4	0.35	0.57	0.08

with respect to a permutation matrix between estimates and references. Then, estimate-reference pairs are discarded that either have an all-zeros reference, or an estimate with power that is 20 dB below the power of the quietest non-zero reference source. To measure how prevalent this is, we compute the proportion of the examples in the following three categories, which are the three right-most columns of table 2:

1. Under-separated: fewer non-zero estimates than non-zero references.
2. Equally-separated: equal number of non-zero estimates and references.
3. Over-separated: more non-zero estimates than non-zero references.

Note that due to the mixture consistency layer (1), the error caused by under-separation is still accounted for, since each under-separated estimate will contain some content of at least one other active non-zero source. From the confusion matrices (Figure 4), the under, equal, and over-separation rates are given by the normalized sums of the lower triangle, diagonal, and upper triangle, respectively.

6. EXPERIMENTAL RESULTS

Table 2 shows the evaluation results in terms of single-source SI-SNR (1S), multi-source SI-SNRi (MSi), and source counting rates for the baseline separation model on both dry and reverberant validation and test sets of FUSS. Note that the model achieves higher scores on the reverberant sets. This is somewhat unexpected, since other separation models in specific domains, such as speech, often perform worse in reverberant conditions [TODO: citation]. We hypothesize that speech is easier to separate based on its reliable spectral characteristics, whereas for arbitrary sounds, with less predictable spectral patterns, models may learn to depend on other cues for separation, such as the differing reverberation pattern for each source. Figure 3 shows scatter plots for input SI-SNR versus separated SI-SNR on dry and reverberant test sets for matched training, which visualize the expected improvement for matched models on both dry and reverberant test sets.

As indicated by the source counting proportions, the baseline model exhibits both under-separation and over-separation, tending to under-separate more often than over-separate, which can be seen in the confusion matrices shown in Figure 4. Incorporating mechanisms or regularizations that help avoid these issues is an interesting avenue of future work. One initial finding in another work [27] is that increasing the number of output sources of the model from 4 to 8, and using mixtures-invariant training (MixIT) [27] helps to avoid under-separation. However, those models seem to suffer more from over-separation, often struggling to reproduce single-source inputs (i.e., achieve low 1S scores), especially if no supervised training examples of single sources are provided.

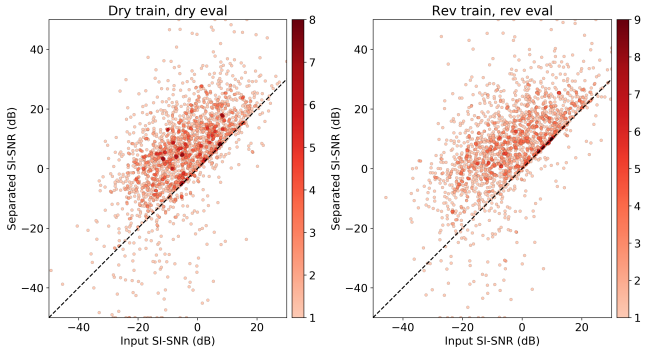

Fig. 3. Scatter plots of model performance for examples with 2-4 sources.

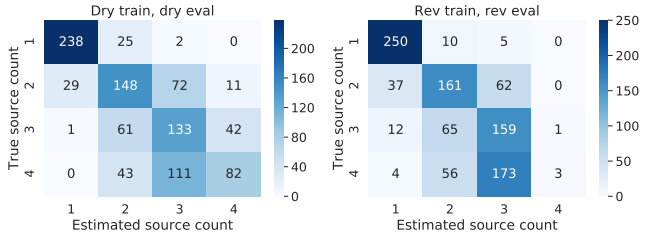

Fig. 4. Confusion matrices for source counting, without reverberation (left) and with reverberation (right).

7. CONCLUSION

We have presented an open-source dataset for universal sound separation with variable numbers of sources, along with a baseline model that achieves surprisingly good performance at this difficult task. Future work will include exploring other mechanisms to avoid under- and over-separation. Incorporating class labels from FSD50K is another interesting avenue of further research. We also plan to release the code for our room simulator and extending its capabilities to address more extensive acoustic properties of rooms, materials with different reflective properties, novel room shapes, and so on.

8. REFERENCES

[1] Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey, "Universal sound separation," in *Proc. WASPAA*. IEEE, 2019, pp. 175–179.

[2] Efthymios Tzinis, Scott Wisdom, John R Hershey, Aren Jansen, and Daniel PW Ellis, "Improving universal sound separation using sound classification," in *Proc. ICASSP*, 2020.

[3] Amir Zadeh, Tianjun Ma, Soujanya Poria, and Louis-Philippe Morency, "Wildmix dataset and spectro-temporal transformer model for monoaural audio source separation," *arXiv preprint arXiv:1911.09783*, 2019.

[4] Efthymios Tzinis, Shrikant Venkataramani, Zhepei Wang, Cem Subakan, and Paris Smaragdis, "Two-step sound source separation: Training on learned latent targets," in *Proc. ICASSP*. IEEE, 2020, pp. 31–35.

[5] Karol J Piczak, "Esc: Dataset for environmental sound classification," in *Proc. ACM Multimedia*, 2015, pp. 1015–1018.

[6] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux, "Finding strength in weakness: Learning to separate sounds with weak supervision," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 2386–2399, 2020.

[7] Qiuqiang Kong, Yuxuan Wang, Xuchen Song, Yin Cao, Wenwu Wang, and Mark D Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *Proc. ICASSP*. IEEE, 2020, pp. 101–105.

[8] Keisuke Kinoshita, Lukas Drude, Marc Delcroix, and Tomohiro Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *Proc. ICASSP*. IEEE, 2018, pp. 5064–5068.

[9] Yi Luo and Nima Mesgarani, "Separating varying numbers of sources with auxiliary autoencoding loss," *arXiv preprint arXiv:2003.12326*, 2020.

[10] Justin Salamon and Juan Pablo Bello, "Feature learning with deep scattering for urban sound analysis," in *Proc. EUSIPCO*. IEEE, 2015, pp. 724–728.

[11] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Proc. EUSIPCO*. IEEE, 2016, pp. 1128–1132.

[12] Emmanouil Benetos, Grégoire Lafay, Mathieu Lagrange, and Mark D Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *Proc. ICASSP*. IEEE, 2016, pp. 6450–6454.

[13] Victor Bisot, Slim Essid, and Gaël Richard, "Overlapping sound event detection with supervised nonnegative matrix factorization," in *Proc. ICASSP*. IEEE, 2017, pp. 31–35.

[14] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen, "Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features," in *Proc. IJCNN*. IEEE, 2018, pp. 1–7.

[15] Toni Heittola, Annamaria Mesaros, Tuomas Virtanen, and Moncef Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Proc. ICASSP*. IEEE, 2013, pp. 8677–8681.

[16] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon, "Sound event detection in synthetic domestic environments," in *Proc. ICASSP*, 2020.

[17] Yuxin Huang, Liwei Lin, Shuo Ma, Xiangdong Wang, Hong Liu, Yueliang Qian, Min Liu, and Kazushige Ouch, "Guided multi-branch learning systems for dcase 2020 task 4," Tech. Rep., DCASE2020 Challenge, June 2020.

[18] Samuele Cornell Cornell, Giovanni Pepe, Emanuele Principi, Manuel Pariente, Michel Olvera, Leonardo Gabrielli, and Stefano Squartini, "The univpm-inria systems for the dcase 2020 task 4," Tech. Rep., DCASE2020 Challenge, June 2020.

[19] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "FSD50K: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[20] Jort F Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[21] Eduardo Fonseca, Jordi Pons, Xavier Favory, Frederic Font, Dmitry Bogdanov, Andrés Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proc. ISMIR*, Suzhou, China, 2017, pp. 486–493.

[22] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. WASPAA*. IEEE, 2017, pp. 344–348.

[23] Nicolas Turpault, Romain Serize, Scott Wisdom, Hakan Erdogan, John R. Hershey, Eduardo Fonseca, Prem Seetharaman, and Justin Salamon, "Improving sound event detection metrics: insights from dcase 2020," submitted to ICASSP 2021.

[24] Zhong-Qiu Wang, Hakan Erdogan, Scott Wisdom, Kevin Wilson, and John R. Hershey, "Sequential multi-frame neural beamforming for speech separation and enhancement," 2020.

[25] Enzo De Sena, Niccolò Antonello, Marc Moonen, and Toon Van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE/ACM TASLP*, vol. 23, no. 4, pp. 774–786, 2015.

[26] Zhong-Qiu Wang, Scott Wisdom, Kevin Wilson, and John R Hershey, "Alternating between spectral and spatial estimation for speech separation and enhancement," *arXiv preprint arXiv:1911.07953*, 2019.

[27] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J. Weiss, Kevin Wilson, and John R. Hershey, "Unsupervised sound separation using mixtures of mixtures," *Advances in Neural Information Processing Systems*, 2020.

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

[29] Scott Wisdom, John R. Hershey, Kevin Wilson, Jeremy Thorpe, Michael Chinen, Brian Patton, and Rif A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. ICASSP*, 2019, pp. 900–904.

[30] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey, "SDR–half-baked or well done?," in *Proc. ICASSP*, 2019, pp. 626–630.

[31] Nicolas Turpault, Scott Wisdom, Hakan Erdogan, John R Hershey, Romain Serizel, Eduardo Fonseca, Prem Seetharaman, and Justin Salamon, "Improving Sound Event Detection In Domestic Environments Using Sound Separation," in *Proc. DCASE Workshop*, 2020.