ALL FOR ONE AND ONE FOR ALL: IMPROVING MUSIC SEPARATION BY BRIDGING NETWORKS

Ryosuke Sawata†*, Stefan Uhlich*‡*, Shusuke Takahashi*† *and Yuki Mitsufuji*†

†Sony Corporation, Tokyo, Japan ‡Sony Europe B.V., Stuttgart, Germany

ABSTRACT

This paper proposes two novel loss functions, a multi domain loss (MDL) and a combination loss (CL), for music source separation with deep neural networks (DNNs). In particular, by using MDL we take advantage of the frequency and time domain representation of audio signals and by using CL we consider the relationship among output source instruments, respectively. MDL and CL can easily be applied to many existing DNN-based methods since they are merely loss functions which are used during training and which do not affect the inference step. Experimental results show that the performance of Open-Unmix (UMX), which is a well-known and state-of-the-art open source library for music source separation, could be improved by utilizing our two new loss functions MDL and CL. Our modifications of UMX with the aforementioned MDL and CL will be made available together with this paper.

Index Terms— Music Source Separation (MSS), Deep Neural Network (DNN), Loss Function

1. INTRODUCTION

In the field of source separation, many approaches have been researched such as local Gaussian modelling [1, 2], non-negative matrix factorization (NMF) [3–5], kernel additive modelling [6] and hybrid methods, which combine these approaches [7, 8]. In particular, there have been many methods which introduce deep neural networks (DNNs) in order to improve the conventional performance in recent years. There are three basic DNN architecture, namely multi-layer perceptrons (MLPs) [9], convolutional neural network (CNNs) [10] and recurrent neural network (RNNs) [11], and all of these have been already introduced for the task of audio source separation. For instance, an MLP was used to separate the input spectra and then obtain separated results in [12, 13]. In [14, 15], CNNs and RNNs were also used to realize source separation which are more successful than previous MLP-based methods since CNNs and RNNs can consider the temporal contexts via convolution and recurrent layers.

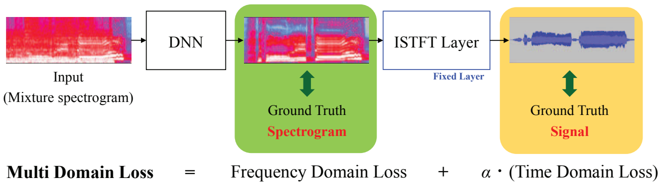
Although the aforementioned researches have improved the performance of conventional source separation drastically, there are two problems with respect to optimization: (1) most existing methods consider only the time or frequency domain and (2) they do not consider the mutual influence among output sources since loss functions are independently applied to each source estimate and the corresponding ground truth instrument. For example, one of the state-of-the-art open-source system for audio source separation, *i.e.*, Open-Unmix (UMX) [16][1], conducted music source separation in only frequency domain since the input and output of UMX are both spectrograms. Furthermore, UMX applied the conventional loss functions to pairs each of which is a masked spectrogram source and the corresponding ground truth independently during training networks. In other words, UMX trains networks one-by-one per each instrument independently by using conventional loss function.

In order to solve the aforementioned problems, we pay attention to the novel schemes with respect to loss functions and bridging networks. In the field of speech enhancement, which is a special case of audio source separation, the methods considering time and frequency domain have been researched in recent years [17, 18]. For instance, Kim *et al.* showed in [17] the effectiveness of multi domain processing via hybrid denoising networks. Furthermore, Su *et al.* reported in [18] that building two discriminators which are responsible for time and frequency domain can realize effective denoising and dereverberation in their scheme of generative adversarial network (GAN). On the other hand, in the field of audio source separation, it is reported the effectiveness of fully-convolutional time-domain audio separation network (Conv-TasNet) [19, 20]. In particular, Défossez *et al.* reported in [20] that the performances of Conv-TasNet was higher than those of UMX, which they were trained and evaluated by using same train and test datasets. One of the reasons is considered that the architecture of Conv-TasNet crosses among sources via channels of convolutional layers while the UMX's one does not cross like Conv-TasNet. Therefore, it is difficult for UMX to consider the mutual influence among source instruments obtained by same input mixture since UMX is consisted by integrating independent each source extraction network.

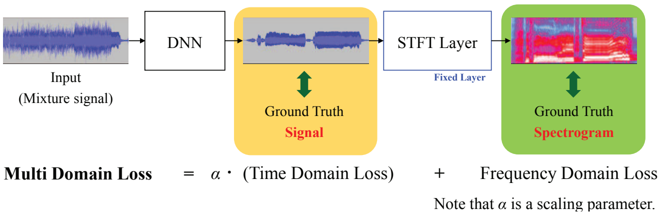
Motivated by the aforementioned, we propose a novel two loss functions and add these to UMX, called UMX+CrossNet (cUMX), in this paper. First is loss function named Multi Domain Loss (MDL). Specifically, we build the additional differentiable short-time Fourier transform (STFT) or inverse STFT (ISTFT) layer by using 1-Dimensional convolution[2] during only training, and then apply loss functions before and after STFT/ISTFT layer. In this way, MDL can consider not only frequency but also time domain differences between input and output spectrograms by applying a loss function in both domains. Second, we also propose the novel loss function, named Combination Loss (CL), with bridging network paths in UMX. As we above mentioned, not only UMX but also almost all conventional methods for source separation train their networks for each source independently. Thus, it is difficult to find a cause source degrading the performance, *i.e.*, what is the kind of sources mixed into the results as noise. To tackle this problem, CL consider the relationship among output sources by generating the output spectrogram combinations and applying the MDL to these. In addition, we bridge the UMX's network paths to cross and share the mutual influence of all source instruments. If the performance of ith source separation is insufficient, the combinations including

[1]https://github.com/sigsep/open-unmix-pytorch

[2]If the type of output is spectrogram, we adopt the ISTFT layer. Meanwhile, we adopt the STFT layer if the type of output is time signal.



(a) Applying frequency domain network with post ISTFT layer.



(b) Applying time domain network with post STFT layer.

Fig. 1: Multi domain loss (MDL)

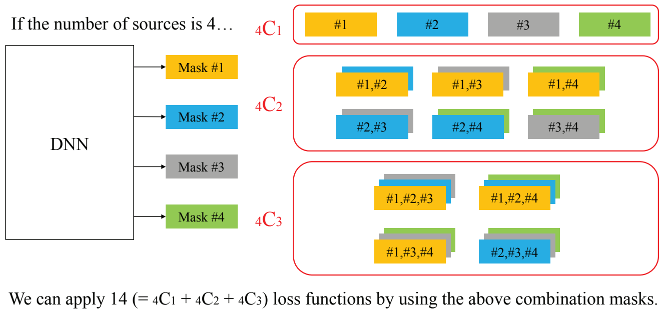


Fig. 2: Combination Loss (CL). To simplify, this figure is an example that the mixture input is consisted of 4 sources.

ith source are adversely affected and the others which do not include ith source are not. Note that bridging operation is specialized handler for a network like UMX since UMX consists of each extraction network independently, not separation network. Namely, a network like Conv-TasNet which is already crossing among sources via convolutional layer does not need like this operation. As a result of this, it is expected to train networks effectively since finding a cause source is possible during training. The above proposals, *i.e.*, MDL and CL, only affect training step and can be introduced to the almost all DNN-based conventional methods since they are just loss functions. In other words, there is no additional learning parameters. Consequently, effective performance improvement according to most DNN-based source separation methods becomes feasible without additional calculating cost at inference time.

2. NOVEL LOSS FUNCTIONS

In this section, we describe the details of our novel loss functions, *i.e.*, MDL and CL, and discuss their expected merits. At first, we assume that the time-domain mixture signal $\boldsymbol{y}$ which is used as DNN input consists of J sources, i.e.,

$$\boldsymbol{y} = \sum_{j=1}^{J} \boldsymbol{x}_j, \tag{1}$$

where $\boldsymbol{x}_j$ denotes the time-domain signal of the jth source. In this section, we also assume that the output of DNN is a mask $\boldsymbol{M}_j$ which can extract a jth desired source from the mixture spectrum $\boldsymbol{Y} = \mathcal{S}\{\boldsymbol{y}\}$:

$$\hat{\boldsymbol{x}}_j = \mathcal{S}^{-1}\{\hat{\boldsymbol{X}}_j\}, \tag{2}$$

$$\hat{\boldsymbol{X}}_j = \boldsymbol{M}_j \circ \boldsymbol{Y}, \tag{3}$$

where $\mathcal{S}$ and $\mathcal{S}^{-1}$ are respectively forward and inverse operators of the STFT. Furthermore, $\hat{\boldsymbol{x}}_j$ and $\hat{\boldsymbol{X}}_j$ are the predicted results of time and frequency domain ground truths, *i.e.*, $\boldsymbol{x}_j$ and $\boldsymbol{X}_j$.

2.1. Multi Domain Loss (MDL)

In the scheme of MDL, we firstly build the additional differentiable and fixed STFT or ISTFT layer after the output layer by using 1-Dimensional convolution. Then we applied the time and frequency domain loss functions before and after STFT or ISTFT layer. Note that the STFT or ISTFT layer does not affect the inference step since this layer is only used during training in order to compute the MDL. In particular, we used the mean squared error (MSE) between separated and ground truth spectrograms as frequency domain loss, and weighted Signal-to-Distortion Ratio (wSDR) [21] as time domain loss. Therefore, in our method MDL of jth source is calculated as follows:

$$\mathcal{L}^j_{\text{MDL}} = \mathcal{L}^j_{\text{MSE}} + \alpha \mathcal{L}^j_{\text{wSDR}}, \tag{4}$$

where α is a scaling parameter. Furthermore, jth source's MSE $\mathcal{L}^j_{\text{MSE}}$ and wSDR $\mathcal{L}^j_{\text{wSDR}}$ are respectively calculated as follows:

$$\mathcal{L}^j_{\text{MSE}} = \sum_{j=1}^{J} \sum_{t,f} \left\{ \boldsymbol{X}_j(t,f) - \hat{\boldsymbol{X}}_j(t,f) \right\}^2, \tag{5}$$

$$\mathcal{L}^j_{\text{wSDR}} = \sum_{j=1}^{J} \left\{ -\rho_j \frac{\boldsymbol{x}_j^{\mathrm{T}} \hat{\boldsymbol{x}}_j}{\|\boldsymbol{x}_j\| \, \|\hat{\boldsymbol{x}}_j\|} - (1-\rho_j) \frac{(\boldsymbol{y}_j - \boldsymbol{x}_j)^{\mathrm{T}} (\boldsymbol{y}_j - \hat{\boldsymbol{x}}_j)}{\|\boldsymbol{y}_j - \boldsymbol{x}_j\| \, \|\boldsymbol{y}_j - \hat{\boldsymbol{x}}_j\|} \right\}, \tag{6}$$

where t and f denote the frame index and frequency bin in the spectrogram, respectively. Moreover, ρ_j is the energy ratio between jth source and mixture, *i.e.*, $\|\boldsymbol{x}_j\|^2 / (\|\boldsymbol{x}_j\|^2 + \|\boldsymbol{y} - \boldsymbol{x}_j\|^2)$.

In summary, the aforementioned scheme is shown in Fig. 1. In this way, taking the advantages of the both of domains becomes feasible via MDL, which can provide improving performance to most conventional methods without requiring additional calculations during inference.

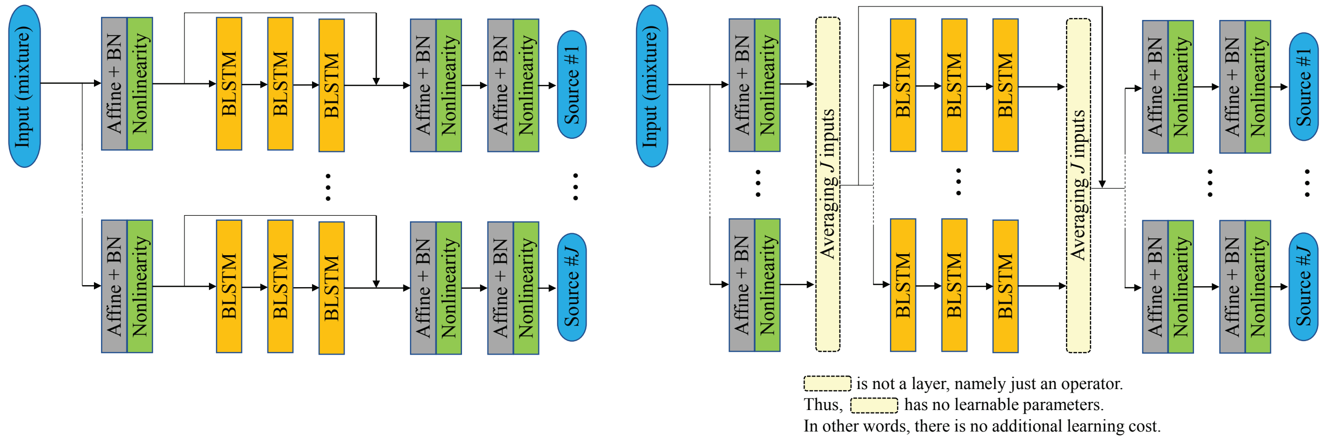
2.2. Combination Loss (CL)

In the scheme of CL, we consider the combinations of output masks. Specifically, we combine the two or more estimated masks into a new those each of which can extract the corresponding two or more sources from input audio. By using the new obtained combination masks, we can apply the more loss functions than when we apply those to pairs each of which is consisted of a separated source and the corresponding ground truth as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}^n_{\text{MDL}}, \tag{7}$$

where $N (> J)$ is the total number of possible combinations, namely $N = \sum_{i=1}^{J-1} {}_J\mathrm{C}_i$, and n denotes the index of nth combination[3]. For

[3]Since we confirmed that the combination ${}_J\mathrm{C}_J$, *i.e.*, the case that output should be equal to input, is not effective in our preliminary experiments, we



(a) UMX

(b) cUMX

Fig. 3: A comparison of network architectures used for our experiments.

example, in the situation of separating four sources, we can consider the 14 (= ${}_4C_1$ + ${}_4C_2$ + ${}_4C_3$) combinations in total although conventional methods consider each source independently, namely the number of masks is 4 (${}_4C_1$) (see Fig. 2).

In order to effectiveness of CL for UMX, we also propose UMX with crossing architecture, named UMX+CrossNet (cUMX). In our method, the aforementioned CL aims to train each network with considering the relationship among output sources by combining output masks. Therefore, we considere that it is necessary to cross not only the loss function via CL but also network graph *i.e.*, differential paths, in order to consider the above relationship. To realize the aforementioned, we connect the paths to cross each source's network by adding just two average operators to original UMX as shown in Fig. 3. Note that this is a special case for UMX since UMX consists of each extraction network independently. Namely, if a target network to apply CL has crossing paths among sources originally, we can use the scheme of CL without like the above operation.

In this way, our method can consider more various sources, *i.e.*, two or more source separation, than considering each source independently via CL. As a result of this, it is expected to train networks effectively since finding a cause source is possible during training. This is because if the performance of ith source separation is insufficient, the combinations including ith source are adversely affected and the others which do not include ith source are not. Furthermore, it is considered that CL can provide the benefit which is similar to scheme of data augmentation due to considering additional two or more sources by using the obtained combination masks.

Consequently, we argue that MDL and CL can provide the performance improvement to most conventional methods without additional calculating cost at inferencing step since they are just loss functions and affect only training step.

3. EXPERIMENTS

3.1. Setup

In our experiments, we evaluated the proposed method on the MUSDB-HQ dataset [22] by using UMX and cUMX. MUSDB18-HQ is comprised of 150 songs each of which is recorded at 44.1kHz sampling rate. It consists of two subsets ('train' and 'test') where we split the train set further into 'train' and 'valid' as defined in the 'musdb' package[4]. For each song, the mixture and its four sources, *i.e., bass, drums, other* and *vocals*, are available and the task is to separate the mixture into the four original sources.

In our method, CL needs that the loss functions should be applied to the whole network integratedly due to considering the combination of output masks each of which is extracted from the same input mixture, which this is a different manner from the original UMX's one. Specifically, original UMX independently builds and trains a network for each source, namely using four optimizers which are respectively responsible for *bass, drums, others* and *vocals*. Therefore, in our experiments, we used only one optimizer per each method although original UMX employed it per each source instrument, namely original UMX has four optimizers in total. In addition, the original UMX has independent networks for each source, and thus applying early stopping imply that it is possible to select the best epoch per each source effectively. However, in our CL scheme, all network parts for each source must be learned simultaneously to consider the combination masks. Then it is considered that applying early stopping and learning rate drops are not work effectively in our method since they can be affected from the source having fast or slow convergence. Hence, we did not use the function of early stopping although original UMX library is set to use it. The rest experimental settings followed the manner of original UMX. The training was conducted with Adam [23], with an initial learning rate of 0.001. Furthermore, we used a spectrogram as input, where their sequence of STFT magnitudes is obtained by using a frame size of 4096 samples with 1/4 (= 1024 samples) overlap.

3.2. Results

To evaluate the performance of our method, we used Signal-to-Distortion Ratio (SDR). Specifically, we used the official SiSEC evaluation BSSEval v4 which is available as Python package 'museval' at https://github.com/sigsep/sigsep-mus-eval.

The explanation of comparative and proposed methods and ex-

exclude it.

[4]`https://github.com/sigsep/sigsep-mus-db/blob/master/musdb/configs/mus.yaml`

Table 1: Details of each method in our experiment and their SDR results.

METHOD	Network	Applying: MDL	CL	Median of frames, Median of tracks: Bass	Drums	Other	Vocals	Avg.
C1	UMX	×	×	4.84	5.77	4.16	6.28	5.26
$\overline{\text{C1}}$	cUMX			5.36	5.80	4.25	5.80	5.30
C2	UMX	✓	×	5.19	5.82	4.29	6.18	5.37
$\overline{\text{C2}}$	cUMX			**5.59**	6.03	4.46	**6.71**	5.70
C3	UMX	×	✓	5.27	5.71	4.00	6.08	5.26
$\overline{\text{C3}}$	cUMX			5.28	6.12	4.09	6.35	5.46
C4	UMX	✓	✓	4.98	5.98	4.12	6.30	5.35
P (proposed)	cUMX			5.53	**6.33**	**4.54**	6.50	**5.73**

Table 2: Comparison of cUMX and the other public methods in-terms of SDR.

METHOD	Median of frames, Median of tracks: Bass	Drums	Other	Vocals	Avg.
cUMX (proposed)	5.53	**6.33**	**4.54**	6.50	**5.73**
UMX [16]	5.07	6.04	4.28	6.25	5.41
Conv-TasNet [20]	5.66	6.08	4.37	**6.81**	**5.73**
Meta-TasNet [24]	5.58	5.91	4.19	6.40	5.52
DEMUCS [25]	**5.83**	6.08	4.12	6.29	5.58

perimental results are shown in Table 1. Note that C1 is nearly equal to original UMX since it has same network architecture without MDL and CL. However, C1's performances were inferior to those of original UMX. As we discussed in Sec. 3.1, this is because the effects of early stopping and learning rate drops. Specifically, in our experiments, C1 was optimized by averaging the four functions with only one optimizer, which is responsible for all parameters, in order to equalize the condition to cUMX's one. Thus, it was difficult for C1 to optimize early stopping and learning rate drops for each source's network while it is possible for original UMX. First, the validity of applying MDL was confirmed since the performances of C2 which only MDL is used for outperformed the C1 and even $\overline{\text{C1}}$. On the other hand, by comparing the performances of C1 and C3, we did not confirm the effectiveness of applying only CL alone since the C3's performances were less than or equal to those of C1. However, if there is the condition which network has the crossing architecture (see Fig. 3(b)), the validity of CL was confirmed since the $\overline{\text{C3}}$'s scores are superior to those of $\overline{\text{C1}}$ as denoted in the table. Therefore, conditional CL, *i.e.*, CL with crossing networks, is valid. In particular, the validity of just crossing network architecture was also confirmed since the results having "+Cross" architecture were *i.e.*, $\overline{\text{C1}}$-$\overline{\text{C3}}$ and P, were superior to the corresponding those, *i.e.*, C1-C4. Moreover, applying MDL with the condition which network has the crossing architecture is more effective than applying only MDL alone. This is because Table 1 showed that the performances of the aforementioned conditional MDL ($\overline{\text{C2}}$) outperformed the those of simple MDL (C2) drastically. Hence, we can argue that music source separation should be trained not each source independently but all sources integrately to consider the relationship among output sources as we denoted in Sec. 1. Next, we can confirm that the collaborative using all our novelty, *i.e.*, MDL and CL, with crossing condition, which is denoted as 'P' in Table 1, indicated the best performances among all methods. In particular, the results of our method (P) are superior to not only the other comparative methods denoted as C1-C4 and $\overline{\text{C1}}$-$\overline{\text{C3}}$ but also those of original UMX although P is at a disadvantage in terms of having only 1 optimizer, *i.e.*, "w/ shared optimizer" as shown in Table 1. Furthermore, we can confirm that the proposed method realized successful music source separation by comparing with public state-of-the-art methods as shown in Table 2.

From the above experimental results, we can describe the following contributions:

Multi Domain Loss (MDL)
: Improving the performances of conventional methods becomes feasible by introducing MDL to during training.

Combination Loss (CL)
: Although it is difficult to improve the performance of conventional methods by using CL alone if the network paths are separated for each source, it becomes beneficial given that all networks are trained jointly.

cUMX
: Crossing network paths realized by simple average operators can enhance the degree of improvement of above MDL and CL drastically. In addition, just applying crossing operation alone can also improve the performances of conventional methods if the paths of original target network are separated.

Finally, we argue that our proposal which collaboratively utilizes the above, *i.e.*, MDL and CL with crossing condition, is the most powerful and effective way to improve the performances of conventional DNN-based methods for music source separation.

4. CONCLUSIONS

In this paper, we proposed two novel loss functions called Multi Domain Loss (MDL) and Combination Loss (CL). We showed that MDL and CL are effective and convenient for many DNN-based source separation methods since both are merely loss functions which are used during training and thus do not change the inference step. Hence, it is easy to apply them to many conventional methods. In this paper, we applied MDL and CL to a well-known and state-of-the-art open source library, *i.e.*, Open-Unmix (UMX), by only adding two average operators (UMX+CrossNet, named cUMX) to the model and the improved results showed the validity of our approach.

5. REFERENCES

[1] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Underdetermined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.

[2] D. FitzGerald, A. Liutkus, and R. Badeau, "PROJET — Spatial audio separation using projections," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 36–40.

[3] A. Liutkus, D. Fitzgerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5.

[4] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep nmf for speech separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 66–70.

[5] Y. Mitsufuji, S. Koyama, and H. Saruwatari, "Multichannel blind source separation based on non-negative tensor factorization in wavenumber domain," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 56–60.

[6] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Trans. on Signal Processing*, vol. 62, no. 16, pp. 4298–4310, 2014.

[7] A. Ozerov and C. Fevotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[8] A. Liutkus, D. Fitzgerald, and Z. Rafii, "Scalable audio separation with light kernel additive modelling," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 76–80.

[9] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Defense Technical Information Center, 1961.

[10] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[12] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. of 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1748–1752.

[13] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 2135–2139.

[14] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 21–25.

[15] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 261–265.

[16] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - A reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, pp. 1667, 09 2019.

[17] J.-H. Kim, J. Yoo, S. Chun, A. Kim, and J.-W. Ha, "Multi-domain processing via hybrid denoising networks for speech enhancement," *arXiv*, 2018.

[18] J. Su, Z. Jin, and A. Finkelstein, "HiFi-GAN: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks," in *Proc. of Interspeech*, 2020 (accepted for publication).

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

[20] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv*, 2019.

[21] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex U-net," *arXiv*, 2019.

[22] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of 3rd International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2015.

[24] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-learning extractors for music source separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 816–820.

[25] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," *arXiv*, 2019.