This paper proposes several improvements for music separation with deep neural networks (DNNs), namely a multi-domain loss (MDL) and two combination schemes. First, by using MDL we take advantage of the frequency and time domain representation of audio signals. Next, we utilize the relationship among instruments by jointly considering them. We do this on the one hand by modifying the network architecture and introducing a CrossNet structure. On the other hand, we consider combinations of instrument estimates by using a new combination loss (CL). MDL and CL can easily be applied to many existing DNN-based separation methods as they are merely loss functions which are only used during training and do not affect the inference step. Experimental results show that the performance of Open-Unmix (UMX), a well-known and state-of-the-art open-source library for music separation, can be improved by utilizing our above schemes. Our modifications of UMX are open-sourced together with this paper.

**Index Terms—** Music source separation (MSS), Deep neural network (DNN), Loss function

1. INTRODUCTION

Many approaches have been researched in the field of music separation such as local Gaussian modelling [1,2], non-negative matrix factorization [3–5], kernel additive modelling [6] and hybrid methods combining these approaches [7,8]. In particular, many methods have been investigated which introduce deep neural networks (DNNs) in order to improve the separation performance in recent years. There are three basic DNN architectures, namely multi-layer perceptrons (MLPs) [9], convolutional neural networks (CNNs) [10] and recurrent neural networks (RNNs) [11], and all of them 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 estimation. In other words, UMX trains networks one-by-one for each instrument.

In order to solve these problems, we propose schemes with respect to the used loss function and model architecture. In the field of speech enhancement, which is a special case of audio source separation, 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 using generative adversarial networks (GANs). Inspired by these reports, we believe that considering both, time and frequency domain, is important to realize effective music source separation. Next, in the field of audio source separation, the effectiveness of Conv-TasNet [19,20], a fully-convolutional time-domain audio separation network, was reported. In particular, Défossez et al. reported in [20] that the performance of Conv-TasNet was higher than the one of UMX when trained and evaluated on the same dataset. One of the reasons in our opinion is that the architecture of Conv-TasNet allows information sharing among sources as the convolutional layers consider all input channels when computing one output channel. In contrast, UMX does not allow such information sharing as it trains independent source extraction networks for each instrument. Therefore, it is difficult for UMX to consider the mutual influence among instruments obtained from the same input mixture.

Motivated by this discussion, we propose two new loss functions and a new model architecture which we add to UMX, called CrossNet-UMX (X-UMX)\(^1\). First, we introduce a multi-domain loss (MDL) where we append an additional differentiable short-time Fourier transform (STFT) or inverse STFT (ISTFT) layer\(^2\) during training only. The loss is computed from the estimates before and after the STFT/ISTFT layer. In this way, MDL can consider not only frequency but also time domain differences between estimates and ground truth. Second, we also propose a further loss function, named combination loss (CL), and bridging network paths for UMX. As we mentioned above, not only UMX but also almost all conventional methods for music source separation train their networks for each source independently. Thus, it is difficult to find the right-cause of performance degradation, i.e., which sources are leaking and thus producing incorrect instrument estimates. To tackle

\(^1\)https://github.com/sigsep/open-unmix- [nabla]pytorch

\(^2\)Our implementation for Nnabla, Sony’s deep learning framework, is available at https://github.com/sony/ai-research-code/tree/master/x-umx. Furthermore, a PyTorch version building upon Asteroid [21] is available at https://github.com/asteroid-team/asteroid/tree/master/egs/musdb18/X-UMX.

\(^3\)If the network outputs a spectrogram, we append an ISTFT layer whereas a STFT layer is added if the network output is a time signal.
2. PROPOSED LOSS FUNCTIONS

In this section, we describe in detail our new loss functions, i.e., MDL and CL, and discuss their merits. We assume that the time-domain mixture signal $x$ consists of $J$ sources, i.e.,

$$x = \sum_{j=1}^{J} y_j,$$

where $y_j$ denotes the time-domain signal of the $j$th source and $x$, $y_j$ are column vectors with the samples. Furthermore, we assume that the output of the DNN is a mask $M_j$ which can extract the $j$th desired source from the mixture spectrum $X = S(x)$:

$$\hat{y}_j = S^{-1}(\hat{Y}_j) = S^{-1}(M_j \circ X),$$

where $S$ and $S^{-1}$ are the forward and inverse operators of the STFT, respectively. Furthermore, $\hat{y}_j$ and $\hat{Y}_j$ are the predicted results of time and frequency domain ground truths $y_j$ and $Y_j$.

2.1. Multi-Domain Loss (MDL)

In the scheme of MDL, we first append an additional differentiable and fixed STFT or ISTFT layer after the output layer as shown in Fig. 1. This allows us to compute the loss in the time as well as the frequency domain. Note that the STFT or ISTFT layer does not affect the inference step since this layer is only used during training for computing the MDL. In our method, we use the mean squared error (MSE) between separated and ground truth spectrograms as frequency domain loss, and the weighted signal-to-distortion ratio (wSDR) [22] as time domain loss, i.e.,

$$L_{\text{MDL}}^x = \alpha \cdot \text{Time Domain Loss} + \text{Frequency Domain Loss}$$

where $\alpha$ is a scaling parameter.

The above proposed loss functions, i.e., MDL and CL, only affect the training step and, therefore, can be introduced to many conventional methods since they are merely loss functions. In addition, our bridging operation requires only a slight network modification but does not increase the number of parameters that need to be learned. Consequently, performance improvements can be gained for most DNN-based source separation methods with introducing almost no additional computational costs at inference time.

2.2. Combination Schemes

In this subsection, we introduce the new combination loss (Sec. 2.2.1) and our new bridging network architecture (Sec. 2.2.2) to help each UMX’s extraction network to support each other.

2.2.1. Combination Loss (CL)

In the scheme of CL, we consider the combinations of output masks. Specifically, we combine two or more estimated masks into new ones where each of them can extract two or more sources from the mixture. By using the newly obtained combination masks, we can compute more loss functions than if we only compare each estimated
model as shown in Fig. 3. Using these average operations does not
addition, we observed that it is effective to cross not only the loss
function via CL but also the network graphs in order to help each
UMX’s extraction network to support each other. Hence, we also
cross other combinations, i.e., \( J \) is equal to the binomial coef-
ficient \( \binom{n}{J} \). For example, in the situation of separating four sources,
we can consider 14 (\( = \binom{4}{1} + \binom{4}{2} + \binom{4}{3} \)) combinations in total as
shown in Fig. 2, while conventional methods consider each source
independently\(^4\).

In order to illustrate the benefit of CL, let us consider the follow-
ing example: Assume that we have a system with leakage of vocals
into drums and other resulting in similar errors that both instruments
exhibit. By considering the combination drums + other, we will notice
that the two errors are correlated, resulting in an even larger
leakage of vocals which we try to mitigate by using our proposed CL
loss.

2.2.2. Bridging Networks
In our method, CL aims to train each network with considering the
relationship among output sources by combining output masks. In
addition, we observed that it is effective to cross not only the loss
function via CL but also the network graphs in order to help each
UMX’s extraction network to support each other. Hence, we also
propose UMX with a crossing architecture, named CrossNet-UMX
(X-UMX). Specifically, we connect the paths to cross each source’s
network by adding just two average operators to the original UMX
model as shown in Fig. 3. Using these average operations does not
change the total numbers of parameters which would not be the case
if we would use a concatenation operation. Please note that this
average operation is only needed for models like UMX since UMX
consists of individual extraction networks.

In this way, our method can consider multiple sources together,
.i.e., two or more source separation, than considering each source independently. From a different viewpoint, CL particularly can be
\(^4\)Initial experiments showed that the combination \( J \) is not adding further performance improvements and, hence, it is not used in Eq. (5).

considered to provide a benefit similar to multi-task learning due to considering multiple sources jointly by computing combinational
masks.

We will see in Sec. 3 that MDL and our combination schemes
provide a performance improvement for UMX. We can expect such
an improvement also for many conventional methods without having to introduce additional computational costs at inference time since
MDL and CL are merely loss functions and the bridging is achieved
with a simple average operation without learnable parameters.

3. EXPERIMENTS
In this section, we conduct music separation experiments in order to
confirm the validity of our method. The task is to separate a song
into its four constituent instruments.

3.1. Setup
In our experiments, we evaluate the proposed method on the
MUSDB18 dataset [23] which is comprised of 150 songs each of
which is recorded at 44.1kHz sampling rate. MUSDB18 consists of
two subsets (‘train’ and ‘test’) where we split the train set further
into ‘train’ and ‘valid’ as defined in the ‘musdb’ package\(^5\). For each
song, the mixture and its four sources, i.e., bass, drums, other and
vocals, are available. As in the original UMX, we operated our
networks in the STFT magnitude domain using a Ham window of
length 4096 with 75% window overlap.

Since CL needs to be applied to the joint instrument network
due to the usage of the combinations of the output masks, we cannot
use the original UMX implementation, which independently builds
and trains a network for each instrument. Hence, we always train in
the following the four separation networks jointly, even in the case
that no combination scheme is used, and the loss function is merely
the mean of the four individual losses for each instrument. This
has the effect that the early stopping at the epoch with the smallest
validation error is not done for each instrument individually (as is
the case for the original UMX) but the early stopping is done at the
same epoch for all four networks. Furthermore, the learning rate
drops, which are determined by the ‘ReduceLROnPlateau’ function
are done at the same epoch. Hence, the results that we obtain in

\(^5\)https://github.com/sigsep/sigsep-mus-db/blob/master/ musdb/configs/mus.yaml
Table 1: Comparison of X-UMX with other public methods in terms of SDR ("median of frames, median of tracks").

| Method         | Bass   | Drums  | Other   | Vocals  | Avg.   |
|----------------|--------|--------|---------|---------|--------|
| UMX [16]       | 5.07   | 6.04   | 4.28    | 6.09    | 5.18   |
| Meta-TasNet [25]| 5.58   | 5.91   | 4.19    | 6.40    | 5.52   |
| Demucs [26]    | 6.83   | 6.08   | 4.12    | 6.29    | 5.58   |
| Conv-TasNet [20]| 5.66   | 6.08   | 4.37    | 6.81    | 5.73   |
| X-UMX (proposed)| 5.43   | 6.47   | 4.64    | 6.61    | 5.79   |

To evaluate the performance of our method, we used the signal-to-distortion ratio (SDR) and sources-to-artifacts ratio (SAR) computed with BSSEval v4 called ‘museval’

Fig. 4: Experimental results for proposed methods.

add up by confirming the differences of the average score as follows: C2 – C1 = 0.22dB, C4 – C1 = 0.37dB. Then the sum of them, i.e., 0.22dB + 0.37dB = 0.59dB, is nearly equal to the improvement from C1 to C6 (= 0.60dB). Meanwhile, we can confirm that only CL seems to be not always provide improvement in terms of SDR when it is used with another scheme by focusing on the values of C4 (w/ Bridging) and C7 (w/ Bridging and CL). On the other hand, we can observe that SAR is always improved by adding CL as shown in Fig. 4(b).

Finally, we can confirm that using all our proposed modifications jointly, i.e., MDL, CL and bridging operation, which is denoted as ‘P’ in Fig. 4, gives the best performance in terms of SDR and SAR among all methods. In addition, we can compare our method to other music source separation public state-of-the-art systems shown in Table 1.
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 nnmf 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 wavefield domain,” arXiv, 2019.

[21] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-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.

[22] 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.

[23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bitner, “The MUSDB18 corpus for music separation,” Dec. 2017.

[24] F.-R. Stöter, A. Liutkus, and N. Ito, “The 2018 signal separation evaluation campaign,” in International Conference on Latent Variable Analysis and Signal Separation. Springer, 2018, pp. 293–305.

[25] 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.

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

[27] D. Ward, H. Wierstorf, R. D. Mason, E. M. Grais, and M. D. Plumley, “Bss eval or peass? predicting the perception of singing-voice separation,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 596–600.