IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023EDP7044

Publicized:2024/04/05

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

MDX-Mixer: Music Demixing by Leveraging Source Signals Separated by Existing Demixing Models

Tomoyasu NAKANO[†a)], ***Nonmember and*** **Masataka GOTO**[†b)], ***Fellow***

SUMMARY This paper presents MDX-Mixer, which improves music demixing (MDX) performance by leveraging source signals separated by multiple existing MDX models. Deep-learning-based MDX models have improved their separation performances year by year for four kinds of sound sources: "vocals", "drums", "bass", and "other". Our research question is whether mixing (*i.e.,* weighted sum) the signals separated by state-of-the-art MDX models can obtain either the best of everything or higher separation performance. Previously, in singing voice separation and MDX, there have been studies in which separated signals of the same sound source are mixed with each other using time-invariant or time-varying positive mixing weights. In contrast to those, this study is novel in that it allows for negative weights as well and performs time-varying mixing using all of the separated source signals and the music acoustic signal before separation. The time-varying weights are estimated by modeling the music acoustic signals and their separated signals by dividing them into short segments. In this paper we propose two new systems: one that estimates time-invariant weights using 1x1 convolution, and one that estimates time-varying weights by applying the MLP-Mixer layer proposed in the computer vision field to each segment. The latter model is called *MDX-Mixer*. Their performances were evaluated based on the source-to-distortion ratio (SDR) using the well-known MUSDB18-HQ dataset. The results show that the MDX-Mixer achieved higher SDR than the separated signals given by three state-of-the-art MDX models.

key words: *Music demixing, Music source separation, 1x1 convolution, MLP-Mixer layer, Time-varying mixing*

1. Introduction

The challenge of Music Demixing (MDX), or Music Source Separation (MSS), is to separate individual source signals such as vocals, drums, and bass from a real-world music acoustic signal. High-performance MDX is an essential technology for a variety of applications that analyze and exploit the characteristics of individual sound sources. In fact, MDX was used to add effects to individual source (instrument) sounds for music appreciation [1] and adjust their volume [1–4], to improve the cochlear implant user's musical experience by adjusting the volume of preferred instruments [5], to synthesize singing voices [6], to acquire feature expressions of singing voices [7], to identify singers [8], to estimate compatibility between singing voices and accompaniment [9], and so on. In addition, there are examples used in commercial software related to music production (e.g., Audionamix XTRAX STEMS, `https://audionamix.com/xtrax-stems/`).

[†]The authors are with National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba-shi, 305-8568 Japan.
a) E-mail: t.nakano@aist.go.jp
b) E-mail: m.goto@aist.go.jp

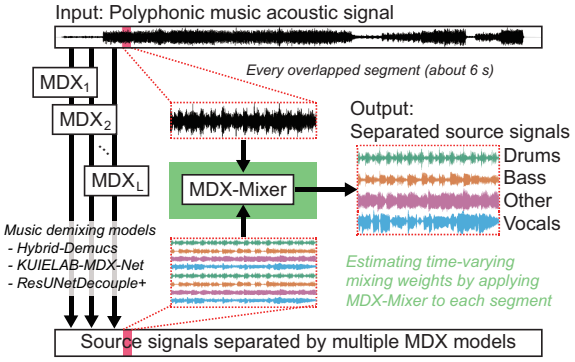

Fig. 1 MDX-Mixer overview. The music acoustic signal and source signals separated by several existing MDX models are mixed approximately every 6 seconds to obtain the final separated signal.

To build a higher-performance MDX framework with better generalization performance, researchers have been working on developing model architectures and training methods and have been preparing and augmenting the diversity of a vast amount of training data. As for model architectures, Deep Neural Networks (DNNs) are widely used as one of the best performing frameworks [10, 11], and current deep MDX models can be broadly classified into the following four types:

(1) separation in the amplitude spectral domain [12–21],
(2) separation in the complex spectral domain [22–25],
(3) separation in the waveform domain [26–30],
(4) hybrid separation of waveforms and complex spectra [31–33].

Other areas of research include increasing the amount and diversity of training data [24, 34, 35], dealing with small numbers of training data by using few-shot learning [36], and estimating synthesis parameters for musical instrument from sound mixtures [37].

In most MDX studies, source-to-distortion ratios (SDRs) of separation methods are evaluated and compared for four types of sources (Drums, Bass, Other, and Vocals). Table 1 shows the SDRs of the four state-of-the-art (SOTA) models with the highest SDRs for each source as of February 2023[†]. The top two rows show the SOTA model for MUSDB18 [38], and the bottom two rows show the SOTA model for MUSDB18-HQ [39] (the frequency-bandwidth-widened version of MUSDB18).

[†]Models published in peer-reviewed conferences are listed.

Table 1 SDRs in the two datasets MUSDB18 and MUSDB18-HQ for the SOTA models of MDX, where "All" means the average of the four source results. **Bold** font denotes the maximum value in each dataset. Models marked with "†" were evaluated by the median SDR in all frames of all 50 songs; the others were evaluated by the "median of frames, median of tracks".

Model			Test SDR in dB				
ID	Name	Dataset	All	Drums	Bass	Other	Vocals
A	CDE-HTCN [33]	MUSDB18	**6.89**	**7.33**	**7.92**	4.92	7.37
B	ResUNetDecouple+† [25]	MUSDB18	6.73	6.62	6.04	**5.29**	**8.98**
C	KUIELAB-MDX-Net† [31]	MUSDB18-HQ	7.54	7.33	7.86	**5.95**	**9.00**
D	Hybrid-Demucs [32]	MUSDB18-HQ	**7.68**	**8.24**	**8.76**	5.59	8.13

Table 1 shows that there are different models for obtaining the best SDR for each sound source. In other words, the research question "Can the best performance be obtained or exceeded for all four sources by utilizing the source signals separated by multiple MDX models?" can be considered. In fact, there are two MDX studies [31, 40] that mix separated source signals in a time-invariant manner to improve separation performance. In those studies, single source signals separated by two different models (*e.g.,* two separated vocal signals) were mixed using positive weights.

Furthermore, since the optimal weights of the mixing may change from time to time depending on the music content, a method was proposed to estimate time-varying positive mixing weights only for the singing voice separation, and its effectiveness was reported [41]. However, the three sound sources other than the singing voice (*i.e.,* Drums, Bass, and Other) were never evaluated.

Unlike these previous studies, this paper proposes a system, MDX-Mixer, with the following three novelties.

1. We propose a system for time-varying mixing by using not only the separated source signals, but also the music acoustic signal before separation (hereinafter simply referred to as "music acoustic signals"). As shown in Figure 1, time-varying weights can be obtained by dividing the music acoustic signal and its separated signals (by existing MDX models) into short segments and estimating the mixing weights.
2. MDX-Mixer mixes not only the separated signals of the same sound source but also the separated signals of all four types of sound sources and the music acoustic signal. For example, to obtain a separated singing voice signal, a separated drum signal is also mixed as a sound source other than the singing voice.
3. Negative weights are also allowed in order to take into account utilization of the music acoustic signal and separated signals of different types of sound sources. The negative wights are expected to be effective for the removal of residual signals of different types of sound sources.

2. Related work

Previously, there have been studies called *blending, fusion, ensemble, combine, etc.*, which mix multiple source separation models or their separation results [31, 40–48]. For speech enhancement or speech separation, focusing on the amplitude spectrum, models were integrated or estimated masks were mixed [42, 43, 45, 46].

In the context of MDX or singing-voice/accompaniment-sound separation, there are studies that utilize the results of multiple separation methods as input to another model. For singing-voice/accompaniment-sound separation, McVicar *et al.* [47] proposed a method to estimate an amplitude spectral mask by a conditional stochastic field, using the outputs from multiple source separation methods as feature vectors.

In addition, there are studies that use multiple separation methods as one of the components of another model. Driedger *et al.* [44] proposed a multi-stage system consisting of multiple separation methods for the amplitude spectrum, focusing on different properties such as harmonic and percussive components.

Moreover, there are studies that are closely related to this paper and that select [48] or mix [31, 40, 41] separated signals. Manilow *et al.* [48] proposed a method to train a DNN model that estimates the SDRs of multiple separation methods every short time and selects the separation results to maximize the predicted SDR. Uhlich *et al.* [40] and Kim *et al.* [31] mixed the separated source signals by using time-invariant positive mixing weights.

Let $x_{i,\text{model1}}(t)$ and $x_{i,\text{model2}}(t)$ be the source signals separated by two MDX models (model1 and model2). They are mixed as follows using the time-invariant weights $w_i(t)$ for each source i:

$$\hat{x}_i(t) = w_i x_{i,\text{model1}}(t) + (1 - w_i) x_{i,\text{model2}}(t). \quad (1)$$

Uhlich *et al.* [40] determined a time-invariant source-independent weight w_i that maximizes the average SDR for the DSD100 Dev set. The optimal $w_i = 0.25$ determined in that way was used to mix the signals separated by the feed-forward model (model1) and the BLSTM model (model2). Kim *et al.* [31] used source-dependent weights w_i to mix the signals separated by a modified TFC-TDF-U-Net [22] (model1) and Demucs [49] (model2). Specifically, w_i was set to 0.5, 0.5, 0.7, and 0.9 for bass, drums, other, and vocals in MDX Challenge 2021 [50]†.

As for time-varying mixing of separated signals, there is a work by Jaureguiberry *et al.* [41] for singing voice separation. This is a method to estimate positive mixing weights

†https://github.com/kuielab/mdx-net/blob/Leaderboard_A/README_SUBMISSION.md

$\sum_i w_{i,n} = 1$ conditional on $\forall w_{i,n} \geq 0$ at time n, using as input the short-time power spectra of the music acoustic signal and multiple separated singing signals.

However, the effectiveness of time-varying mixing weights has never been evaluated for the mixing of separated signals other than singing voices, nor have music acoustic signals or other types of sound sources been utilized as targets for negative mixing weights.

3. Method

In order to mix the music acoustic signal and separated source signals (hereafter simply referred to as the separated signals) in the waveform domain, this paper proposes two new systems: (1) a comparison system that learns time-invariant weights using 1x1 convolution, and (2) MDX-Mixer that estimates time-varying weights using the MLP-Mixer layer [51].

In this paper, we define $\mathbf{X}$ as the input signal that consists of both a stereo music acoustic signal[†] and a multi-channel (C-channel) signals separated by several existing MDX models. The input signal $\mathbf{X}$ is segmented into short segments of T samples, and the kth segment is denoted as $\mathbf{X}_k \in \mathbb{R}^{T\times(2+C)}$. In other words, the $\mathbf{X}_k$ is obtained by concatenating the matrix of 2-channel music acoustic signals of T samples and the matrix of C-channel separated signals of T samples. If four stereo sources separated by one MDX model and a stereo source signal separated by another MDX model are used, then $C = 10$. Each MDX model does not necessarily have to output all 4 target sound sources. For example, a MDX model that outputs only vocals can be used.

The comparison system uses the same (time-invariant) weight $\mathbf{W}$ for each segment to obtain the separated signal $\mathbf{Y}_k = \mathbf{W}\mathbf{X}_k$. In contrast, the proposed MDX-Mixer obtains the separated signal $\mathbf{Y}_k = \mathbf{W}_k\mathbf{X}_k$ by using different (time-varying) weights $\mathbf{W}_k$ for each segment.

The separated signal $\mathbf{Y}_k$ is an 8-channel signal consisting of stereo signals for four sound sources: "Drums", "Bass", "Other", and "Vocals".

3.1 1x1 convolution: Time-invariant mixing

This paper uses 1x1 convolution to mix the music acoustic signal with the source signals separated by "multiple" MDX models. This is for achieving the stereo separated signal $\mathbf{Y}_k$ by removing residual signals of different types of sound sources and by enhancing the targeted sound source signal. In contrast, previously, Kim *et al.* [31] used 1x1 convolution on the music acoustic signal and the signal separated by a "single" MDX model to remove residual signals from other sources.

Our 1x1 convolution-based system is a special case of our proposed MDX-Mixer as shown in Figure 2 and estimates a time-invariant weight matrix $\mathbf{W}$. The system estimates

[†]The left (L) and right (R) channels of each signal are treated as independent signals, so the stereo signal is 2-channel.

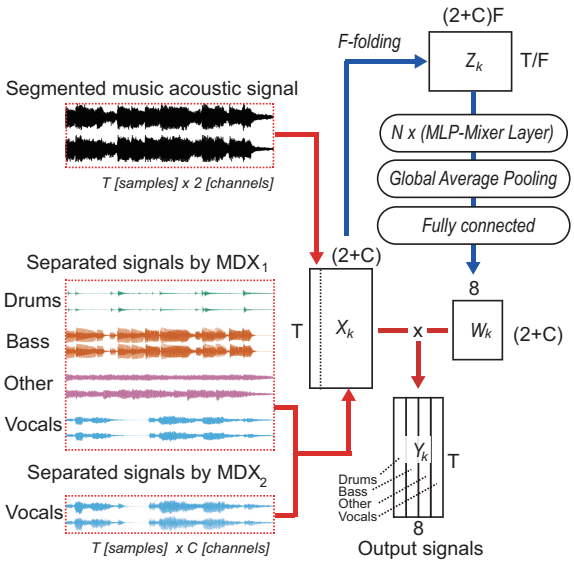

Fig. 2 MDX-Mixer system overview. The system estimates time-varying mixing weight matrix $\mathbf{W}_k$. If the process indicated by the blue arrow is not performed, the time-invariant weights are estimated by 1x1 convolution, in which case the weight matrix is called $\mathbf{W}$.

$\mathbf{W} \in \mathbb{R}^{(2+C)\times 8}$ taking the inner product $\mathbf{Y}_k = \mathbf{X}_k\mathbf{W}$ with $\mathbf{X}_k$ to obtain the 8-channel separated signal $\mathbf{Y}_k$. Since $\mathbf{W}$ is the same weight regardless of the input signal $\mathbf{X}_k$, it expresses the degree of influence of each MDX model on each target source. The 1x1 convolution-based system thus considers inter-channel (inter-source) relationships.

3.2 MDX-Mixer: Time-varying mixing

Figure 2 shows an overview of the MDX-Mixer system. The input signal $\mathbf{X}$ is segmented to obtain $\mathbf{X}_k$, which is then mixed with (multiplied by) the weights $\mathbf{W}_k$ to estimate $\mathbf{Y}_k$. This segmentation allows the time-varying content of $\mathbf{X}$ to be taken into account. The rows of the matrix $\mathbf{X}_k$ represent time and the columns represent sources (channels). As described in Section 3.1, the 1x1 convolution-based system takes into account the inter-channel (inter-source) relationship. In contrast, MDX-Mixer can consider the intra-channel relationship between different times (*i.e.,* different samples within a segment) in addition to the inter-channel relationship.

Therefore, as an extension of the 1x1 convolution, we use the MLP-Mixer layer [51] that can be expressed in terms of full connections between channels and can also consider full connections within channels. The MLP-Mixer layer has been proposed in the computer vision field and has the advantages of simple structure, high performance, low training cost, and high inference throughput. Through this MLP-Mixer layer, we estimate the time-varying mixing weights $\mathbf{W}_k \in \mathbb{R}^{(2+C)\times 8}$ and obtain the 8-channel output signal $\mathbf{Y}_k$ of the four stereo sources as the product of $\mathbf{X}_k$ and $\mathbf{W}_k$ (*i.e.,* $\mathbf{Y}_k = \mathbf{X}_k\mathbf{W}_k$).

The architecture of the MLP-Mixer layer is shown in

Figure 3. In [51], the input image is divided into patches and used as a multi-channel signal, and the image class is estimated by repeating *token-mixing MLP*, which is the fully connected MLP within each divided image (channel), and *channel-mixing MLP*, which is the fully connected MLP between all divided images (channels). If T samples and $(2+C)$-channel matrices are used as input, the size of the weight matrix $\mathbf{W}_{\text{token}}$ required for token-mixing MLP becomes huge when T is large. To reduce its size, T samples are divided (folded) by F and concatenated in the channel direction to obtain the matrix $\mathbf{Z}_k \in \mathbb{R}^{T/F\times(2+C)F}$.

In our current implementation, $T = 2^{18}$ (about 6 seconds with the sampling frequency of 44.1 kHz) is used. Assuming the number of channels to be $(2 + C) = 12$, the size of the weight matrix W_{token} of the token-mixing MLP without folding is $(2^{18})^2$, and the weight matrix W_{channel} has a size of 12^2. Folding them with $F = 2^8$ reduces their sizes to $(2^{10})^2$ and $(12 \times (2^8))^2$, respectively, which are 0.000153 times smaller. Such a folding results in token-mixing MLP modeling the relationships within folded patches and channel-mixing MLP modeling the relationships between those patches.

As shown in Figure 2, the F-folded matrix $\mathbf{Z}_k$ passes through N MLP-Mixer layers. Each MLP-Mixer layer consists of a Layer Normalization [52], skip connections, a token-mixing MLP, and a channel-mixing MLP (Figure 3). The token-mixing MLP and channel-mixing MLP have Gaussian Error Linear Unit (GELU) [53], dropout, and fully connected layers. Since both the token-mixing MLP and channel-mixing MLP have two fully-connected layers, respectively, we can design the number of hidden dimensions between the layers, denoted D_T and D_C for time and channel, respectively. The N MLP-Mixer layers are followed by the Global Average Pooling, which averages across channels and reduces the number of elements while summarizing the data, and finally the weight matrix $\mathbf{W}_k$ is obtained through a fully connected layer.

The proposed MDX-Mixer can thus take into account the time-varying content of music by estimating different weights $\mathbf{W}_k$ for different segments $\mathbf{X}_k$. If appropriate time-varying mixing weights could be estimated, they could lead to higher separation performance.

4. Experiment

To evaluate the effectiveness of the proposed 1x1 convolution-based system and MDX-Mixer, these models were trained using the standard MUSDB18-HQ dataset [39]. 86 songs were used as training data, 14 songs as validation data, and 50 songs as test data.

Four sound sources –"drums," "bass," "other," and "vocals"– were used for separation, and the music acoustic signals were stereo with a sampling frequency of 44.1 kHz.

Separation performance was evaluated by calculating SDR using the *museval* Python package[†]. As in most previous studies ([19, 32], etc.), the SDR of each source is calculated by taking the median values over all 1-second segments of each song to obtain the SDR of the track, and then taking the median of all tracks (*i.e.,* "median of frames, median of tracks").

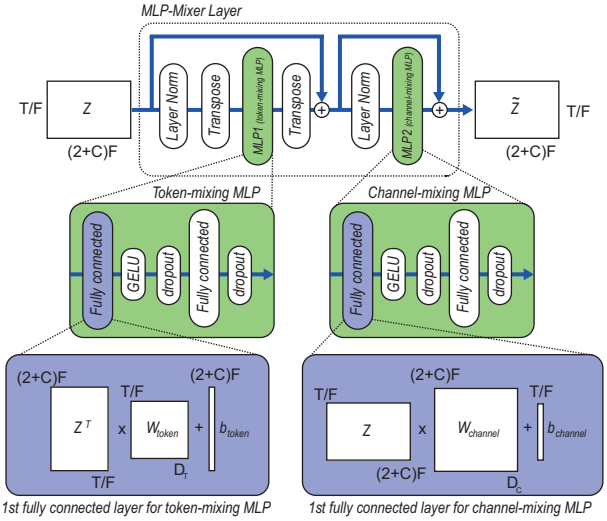

Fig. 3 Overview of MLP-Mixer layer.

4.1 MDX models used for mixing

To focus on discussing performance differences between our approach and existing methods, we used public pre-trained models available on the web for research purposes. Since the pre-trained models for IDs "B", "C", and "D" in Table 1 were publicly available, they were used as existing MDX models as follows to obtain the separated signals for mixing. The pre-trained model for ID "A" was not used here because it was not publicly available.

Model B: "ResUNet143 Subband vocals": A public pre-trained model of ResUNetDecouple+ [25][††]. Its SDR for Vocals is the highest in MUSDB18 and more than 1.6 dB higher than that of another model (Table 1). Since only the vocal/accompaniment separation model is publicly available, only the vocal separated signal was used.

Model C: "kuielab_mdxnet_A": A public pre-trained model of KUEILAB-MDX-Net [31][†††]. Highest SDR for Vocals and Other in MUSDB18-HQ (Table 1). The final layer includes the process of mixing the music acoustic signal and the four sources with 1x1 convolution.

Model D: "mdx" A public pre-trained model of Hybrid-Demucs [32][††††]. The highest SDRs for drums and

[†] https://github.com/sigsep/sigsep-mus-eval

[††] https://github.com/bytedance/

[†††] https://github.com/kuielab/mdx-net-submission/tree/leaderboard_A

[††††] https://github.com/facebookresearch/demucs

Table 2 SDRs in MUSDB18-HQ for the pre-trained MDX models. Models with "*" are trained on MUSDB18-HQ training data. The highest value for each source is shown in **bold** font.

Model		Test SDR in dB				
ID	Name	All	Drums	Bass	Other	Vocals
B	"ResUNet143 Subband vocals" (ResUNetDecouple+ [25])	N/A	N/A	N/A	N/A	8.21
C	"kuielab_mdxnet_A" (KUIELAB-MDX-Net [31])*	7.47	7.20	7.83	**5.90**	**8.97**
D	"mdx" (Hybrid-Demucs [32])*	**7.77**	**8.21**	**9.28**	5.50	8.10

Table 3 SDRs in MUSDB18-HQ for the system estimating time-invariant mixing weights $\mathbf{W}$ based on 1x1 convolution. The **bold** font means that the value is greater than the highest value in Table 2, and the highest value in this table is indicated by an underline. The notation "*" means that the separated signal for that source was not mixed. Specifically, ID: 1-0 is the condition of not using the existing MDX model (*i.e.,* $C = 0$) and ID: 1-1 is the condition using the MDX model in which only stereo vocal signal is used. For IDs: 1-4, 1-5, and 1-6, the results of three runs with different random seeds under the same conditions are shown.

1x1 convolution				Test SDR in dB				
ID	B	C	D	All	Drums	Bass	Other	Vocals
1-0				0.68	0.36*	0.85*	1.32*	0.20*
1-1	✓			2.82	0.58*	1.13*	1.48*	8.09
1-2		✓		7.36	7.19	7.46	5.98	8.8
1-3			✓	7.73	**8.25**	9.00	5.55	8.11
1-4-1	✓		✓	**7.83**	**8.29**	9.00	5.70	8.31
1-4-2	✓		✓	**7.81**	**8.24**	9.00	5.71	8.30
1-4-3	✓		✓	**7.83**	**8.25**	8.98	5.71	8.37
1-5-1		✓	✓	**8.13**	**8.34**	9.04	**6.18**	8.97
1-5-2		✓	✓	**8.00**	**8.37**	8.87	**6.18**	8.59
1-5-3		✓	✓	**8.04**	**8.39**	9.04	**6.17**	8.55
1-6-1	✓	✓	✓	**8.03**	**8.26**	9.06	**6.23**	8.57
1-6-2	✓	✓	✓	**8.05**	**8.35**	8.99	6.31	8.54
1-6-3	✓	✓	✓	**8.16**	**8.34**	9.08	**6.19**	**9.05**

Table 4 SDRs in MUSDB18-HQ for MDX-Mixer estimating time-varying mixing weights $\mathbf{W}_k$. The **bold** font means that the value is greater than the highest value in Table 2, and the highest value in this table is indicated by an underline. "*" means that no separated signal was given for that source.

MDX-Mixer						Test SDR in dB				
ID	B	C	D	N-layers	dropout p	All	Drums	Bass	Other	Vocals
2-0				8	0	0.77	0.80*	0.68*	1.08*	0.51*
2-1	✓			8	0	2.90	1.13*	0.97*	1.43*	8.06
2-2		✓		8	0	7.45	7.19	7.83	5.87	8.92
2-3			✓	8	0	7.72	**8.25**	8.98	5.54	8.09
2-4	✓		✓	8	0	**8.04**	**8.65**	9.17	5.77	8.59
2-5		✓	✓	8	0	**8.16**	**8.24**	9.22	**6.19**	8.97
2-6	✓	✓	✓	8	0	**8.21**	**8.29**	9.19	**6.26**	**9.08**
2-7	✓		✓	8	0.2	**7.94**	**8.42**	9.18	5.76	8.42
2-8		✓	✓	8	0.2	**8.16**	**8.24**	9.28	**6.27**	8.85
2-9	✓	✓	✓	8	0.2	**8.17**	**8.29**	9.20	**6.22**	8.97
3-0				16	0	0.76	0.81*	0.67*	1.05*	0.52*
3-1	✓			16	0	2.84	1.03*	0.85*	1.41*	8.10
3-2		✓		16	0	7.45	7.19	7.78	**5.93**	8.91
3-3			✓	16	0	7.72	**8.26**	8.97	5.54	8.11
3-4	✓		✓	16	0	**7.96**	**8.40**	9.04	5.80	8.57
3-5		✓	✓	16	0	**8.16**	**8.29**	9.16	**6.16**	**9.02**
3-6	✓	✓	✓	16	0	**8.21**	**8.54**	9.10	**6.17**	**9.01**
3-7	✓		✓	16	0.2	**7.95**	**8.63**	9.12	5.73	8.34
3-8		✓	✓	16	0.2	**8.21**	**8.29**	9.26	**6.25**	**9.03**
3-9	✓	✓	✓	16	0.2	**8.15**	**8.27**	9.20	**6.23**	8.90

bass in MUSDB18-HQ, and also the highest average ("All") of the four sound sources (Table 1).

The evaluation results of these models in MUSDB18-HQ are shown in Table 2. Note that the results are not exactly the same as in Table 1 due to differences in training datasets, models, evaluation methods, etc. Our goal here is to obtain performance beyond these SDRs by mixing different MDX models.

Table 5 The highest SDR value from the existing MDX model for MUSDB18-HQ (Table 2), the average of three runs of the system based on 1x1 convolution (Table 3), and the results from MDX-Mixer with different number of layers N and the average of the results with dropout p (Table 4). For example, "CD" means the mixture of the music acoustic signal and the signal separated by models C and D (ID: 2-5, 2-8, 3-5, 3-8) for MDX-Mixer.

ID	All	Drums	Bass	Other	Vocals
max(Table 2)	7.77	8.21	9.28	5.90	8.97
1x1 convolution					
mean(BD)	7.82	8.26	8.99	5.71	8.33
mean(CD)	8.06	8.37	8.99	6.18	8.70
mean(BCD)	8.08	8.32	9.04	6.24	8.70
MDX-Mixer ($T = 2^{18}$, $F = 2^7$)					
mean(BD)	7.97	8.53	9.13	5.76	8.48
mean(CD)	8.17	8.27	9.23	6.22	8.97
mean(BCD)	8.19	8.35	9.17	6.22	8.99

4.2 Training 1x1 convolution and MDX-Mixer

The proposed systems were trained using only a music acoustic signal or, in addition, separated signals obtained using one or more of the MDX models.

The number of samples T should be set to a power of 2 in order to fold F in the time direction. In this paper, both 1x1 convolution-based system and MDX-Mixer were trained on segments of $T = 2^{18}$ (about 6 seconds) with a shift interval of 2^{15} (about 0.7 seconds).

The hyperparameters specific to the MDX-Mixer were $F = 2^7$, the number of MLP-Mixer layers $N = 8, 16$, and a dropout probability p of 0 or 0.2. The N and p were determined with reference to previous studies [51, 54] that used MLP-Mixer layers. The number of dimensions of the hidden layers D_T and D_C were set to be the same size as $\mathbf{Z}_k$, i.e., $D_T = T/F$ and $D_C = (2 + C)F$.

Both systems were trained using the following L1 loss function $\mathcal{L}$ between separated signals $\mathbf{Y}_k$ and predicted signals $\hat{\mathbf{Y}}_k$.

$$\mathcal{L} = |\hat{\mathbf{Y}}_k - \mathbf{Y}_k| \tag{2}$$

Adam optimizer [55] was used to optimize the model parameters with a learning rate of 0.0003. The training was distributed across multiple GPUs, with a batch size of 4 on each GPU. The parameter to be optimized in 1x1 convolution is $\mathbf{W}$, which can be implemented as an fully connected layer. On the other hand, the parameters to be optimized in MDX-Mixer are the weights of all the multiple fully connected layers in the MLP-Mixer layer shown in Fig. 3 and weights of a fully connected layer shown in Fig. 2.

The 1x1 convolution-based system was trained for 50 epochs. The MDX-Mixer was also trained for 100 epochs under the same conditions. For the system based on 1x1 convolution, the validation loss converged around 20 epochs, while for MDX-Mixer, the loss fluctuated during the 100 epochs but tended to decrease gradually. The waveforms were normalized so that the mean amplitude of the music acoustic signal was 0 and the standard deviation was 1.

Table 6 Average of 4 conditions of SDR in MUSDB18-HQ for MDX-Mixer when the music acoustic signal is not used for mixing. If the value exceeds the value for the same condition in Table 5 where the music acoustic signal is used for mixing, it is indicated by **bold** font. Conversely, if the value decreased, a ↓ is annexed.

ID	All	Drums	Bass	Other	Vocals
MDX-Mixer (without music acoustic signal)					
mean(CD)	8.17	8.27	9.23	6.14↓	**9.05**
mean(BCD)	8.15↓	8.27↓	9.13↓	6.15↓	**9.06**

Table 7 Average of 4 conditions of SDR in MUSDB18-HQ for MDX-Mixer when T and F are changed. If the value exceeds the value for the same condition in Table 5, it is indicated by **bold** font. Conversely, if the value decreased, a ↓ is annexed.

ID	All	Drums	Bass	Other	Vocals
MDX-Mixer ($T = 2^{17}$, $F = 2^6$)					
mean(CD)	**8.19**	**8.44**	9.15↓	6.18↓	**8.99**
mean(BCD)	8.19	**8.44**	9.16↓	6.15↓	**9.04**
MDX-Mixer ($T = 2^{16}$, $F = 2^5$)					
mean(CD)	**8.18**	**8.33**	9.22↓	6.16↓	**9.03**
mean(BCD)	8.19	**8.41**	9.17	6.14↓	**9.06**

For each training condition, the model with the smallest validation loss was used for the test evaluation. In the test data separation, the music acoustic signal and its separated signal obtained by the MDX model were divided into segments $\mathbf{X}_k$ of fixed length T with shift width $T/4$, and their mixed results $\mathbf{Y}_k$ were weighted overlap-added to obtain the final signal $\mathbf{Y}$.

4.3 Results

Tables 3 and 4 show the results of the systems trained by the different hyperparameters. The check marks in columns "B", "C", and "D" indicate the MDX model used to obtain the separated signals. If none of them are checked, it means that only the music acoustic signal was input as X_k, which corresponds to $C = 0$ and $\mathbf{X}_k \in \mathbb{R}^{T\times 2}$. Model B outputs only the vocal separated signal, while the other models output all four source separated signals.

Table 5 shows the average of the results of three runs (training with the same hyperparameters and different random seeds) in the system based on 1x1 convolution. It also shows the average of the results for the MDX-Mixer condition using the same MDX models but with different hyperparameters.

To validate the effectiveness of the systems in more detail, SDR averages for the conditions using models C and D (ID: 2-5, 2-8, 3-5, 3-8) and for the conditions using models B, C, and D (ID: 2-6, 2-9, 3-6, 3-9) without using music acoustic signals for mixing are shown in Table 6. Similarly, SDR averages are shown in Table 7 for the results when the segment length T and the splitting factor F are changed. We used ($T = 2^{17}, F = 2^6$) and ($T = 2^{16}, F = 2^5$) to keep one token size T/F (size of token-mixing MLP) constant.

Finally, examples of the estimation result of time-invariant mixing weights $\mathbf{W}$ and time-varying mixing weights $\mathbf{W}_k$ are shown in Figures 4 and 5, respectively. The

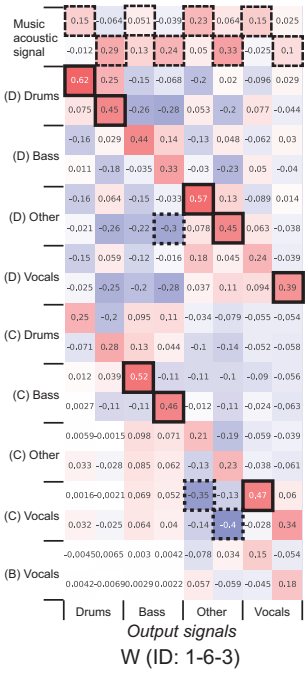

Fig. 4 Example of time-invariant mixing weights $\mathbf{W}$ estimated by the 1x1 convolution-based system. The rows indicate the signals used for the mixing, and (B), (C), and (D) are the IDs of the MDX models.

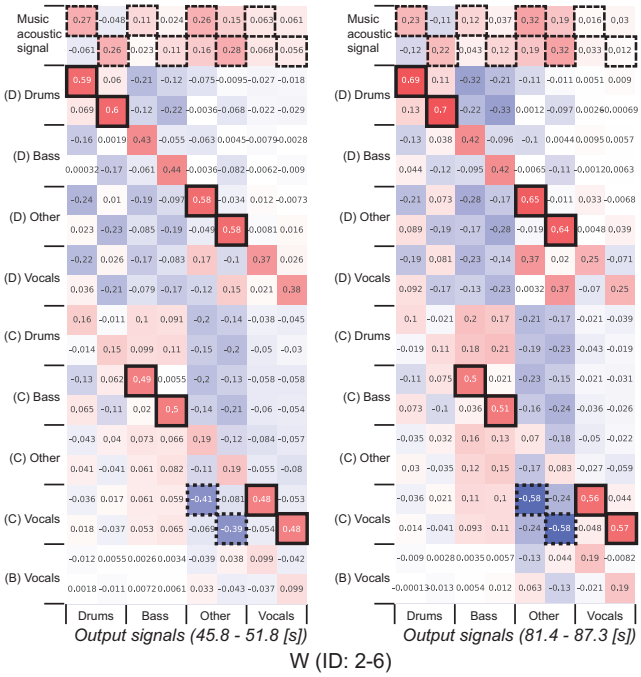

Fig. 5 Examples of time-varying mixing weights $\mathbf{W}_k$ estimated by MDX-Mixer at two different segments (song name: "The Doppler Shift - Atrophy"). Different $\mathbf{W}_k$ were estimated for different segments $\mathbf{X}_k$, i.e., the weights were indeed time-varying.

$\mathbf{W}$ and $\mathbf{W}_k$ were estimated by ID: 1-6-3 and ID: 2-6, respectively, for the MUSDB18-HQ test data "The Doppler Shift - Atrophy". The rows indicate the 20 sources (10 stereo signals) used for the mixing, and (B), (C), and (D) are the IDs of the MDX models. In addition, an example of the mean and standard deviation of $\mathbf{W}_k$ within one song is shown in Figure 6. In Figures 4 to 6, the largest positive weights for each channel of output are marked with bold-line square boxes, large negative weights (less than -0.3) are marked with dotted square boxes, and the weights for the same channel of music acoustic signals are marked with dashed square boxes.

4.4 Discussion

First, the results of applying 1x1 convolution and MDX-Mixer to a single MDX model (ID: 1-/2-/3-1,2,3) show that these SDRs were not improved compared to Table 2 in general. However, a comparison of the results for the condition without the separated signal (ID: 1-0, 2-0, 3-0) with the results for the condition using only the vocal separated signal from Model B (ID: 1-1, 2-1, 3-1) shows that the SDRs for Drums, Bass, and Other were improved. Furthermore, the SDRs for Vocals in the conditions with signals separated by Model B and Model D (ID: 1-4-1, 1-4-2, 1-4-3, 2-4, 2-7, 3-4, 3-7) were better than when Model B and Model D were used alone. This indicates that separation performance could be improved by using other sound sources, as reported by Kim *et al.* [31].

Next, the results of applying 1x1 convolution to multiple MDX models (3 runs each in Table 3: ID: 1-4, 1-5, 1-6) show that it yielded higher performance on average than using a single MDX model did, as shown in the "All" column in Table 5. Here, the BCD condition performed better on average than the BD and CD conditions, indicating that mixing a variety

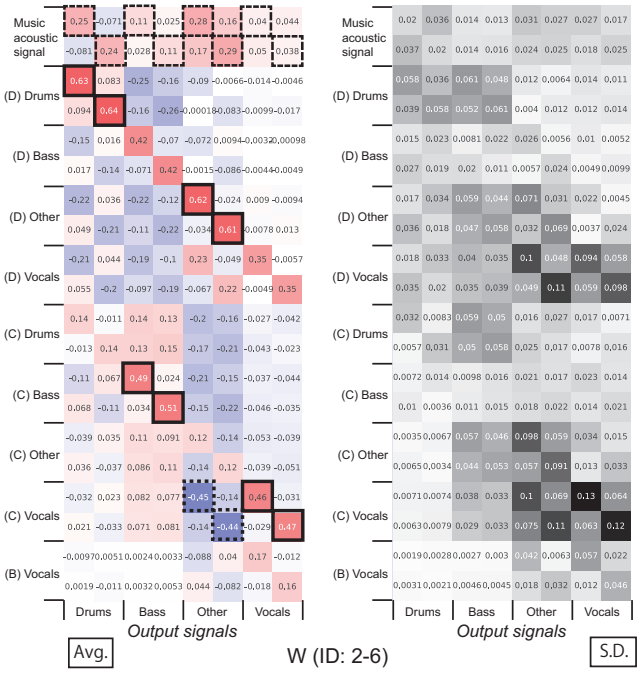

Fig. 6 Example of the mean and standard deviation of $\mathbf{W}_k$ estimated by MDX-Mixer within one song (song name: "The Doppler Shift - Atrophy"). The standard deviations show that the weights changed over time, and in this example, Vocals and Other had particularly large weight variations.

of separated signals is beneficial. And when MDX-Mixer was applied to multiple MDX models (Table 4: ID: 2-4 to 2-9 and 3-4 to 3-9), SDR improved on average compared to the system based on 1x1 convolution (Table 5). The results in Table 4 also show that differences in the number of MLP-Mixer layers N and dropout probability p had little effect on the SDR.

The maximum SDR value for each sound source in Table 2 averages 8.09 dB= $(8.21 + 9.28 + 5.90 + 8.97)/4$. Therefore, as shown in Table 5, the MDX-Mixer can separate sound sources better than manually selecting the MDX model that takes the maximum value for each sound source.

The most improvement occurred when Models C and D were used (ID: 3-8) or when Models B, C, and D were used (ID: 2-6, 3-6), with an "All" of 8.21 dB. This is an improvement of more than 0.44 dB in SDR from the maximum value of 7.77 dB for the MDX model D. These results show the effectiveness of the MDX-Mixer in estimating time-varying mixing weights.

Figures 4 to 6 indicate that the weights of the sources with higher SDRs by each MDX model tended to be larger. Figures 5 and 6 show that the weights changed over time, and in the visualized standard deviations in Figure 6, weights of Vocals and Other showed particularly large changes over time (*i.e.,* had larger standard deviations). Relatively large positive weights were estimated not only for the separated signals but also for the music acoustic signal, indicating that the music acoustic signal was utilized. In fact, if the music acoustic signal was not used for mixing (Table 6), the SDRs for Drums, Bass, and Other decreased on average, while Vocals had a higher average SDR. As shown in the top two rows of Figure 6, the weight of the music acoustic signal was small for Vocals and relatively large for Other, Drums, and Bass, indicating that the music acoustic signal had an influence on the three types of sound sources other than Vocals. Furthermore, negative weights were actually estimated for some music acoustic and separated signals, which was expected to have an effect such as attenuating residual sound from other sound sources. For example, Figures 4 to 6 show that Vocals in Model C had large negative weights for the output of Other, suggesting that separated signals of Vocals were used to remove them from Other.

Finally, changing the segment length T and the segmentation factor F did not change the performance on average from Table 7. However, there was an improvement in Drums and Vocals and a decrease in Bass, so it may be possible to tune the results by adjusting these parameters.

5. Conclusion

This paper proposes two systems that utilize source signals separated by multiple MDX models. The contributions of this paper are as follows.

- We proposed a system using 1x1 convolution that mixes the source signals separated by multiple existing MDX models and the music acoustic signals with time-invariant mixing weights.
- Extending the system based on 1x1 convolution, we also proposed the MDX-Mixer system to estimate time-varying mixing weights.
- We have shown that SDRs can be improved by using multiple existing MDX models for both 1x1 convolution-based system and MDX-Mixer. To answer the research question stated in Section 1, the results show that the MDX-Mixer, which estimates time-varying weights, is superior to the system based on 1x1 convolution and could improve performance over manually selecting an existing MDX model that takes the maximum SDR value for each source.
- Figures 4 to 6 show that the music acoustic signals were utilized by mixing with positive weights. Negative weights were also estimated for the music acoustic signal and the separated signals, which could have been used for removing residual signals of other sound sources, etc.

As more diverse pre-trained models become available in the future, it will become more important to leverage them for target applications. Although this paper showed how to automatically combine pre-trained models to obtain better performance, this approach has potential to be extended to more interactive or semi-automatic ways to combine them according to target applications.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917.

References

[1] J. Woodruff, B. Pardo, and R. Dannenberg, "Remixing stereo music with score-informed source separation," Proc. the 7th International Conference on Music Information Retrieval (ISMIR 2006), pp.314–319, 2006.

[2] O. Gillet and G. Richard, "Extraction and remixing of drum tracks from polyphonic music signals," Proc. the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005), pp.315–318, 2005.

[3] K. Yoshii, M. Goto, and H.G. Okuno, "INTER:D: A drum sound equalizer for controlling volume and timbre of drums," Proc. the 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT 2005), pp.205–212, 2005.

[4] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H.G. Okuno, "Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models," Proc. the 9th International Conference of Music Information Retrieval (ISMIR 2008), pp.133–138, 2008.

[5] J. Pons, J. Janer, T. Rode, and W. Nogueira, "Remixing music using source separation algorithms to improve the musical experience of cochlear implant users," The Journal of the Acoustical Society of America, vol.140, no.6, pp.4338–4349, 2016.

[6] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.Y. Liu, "DeepSinger: Singing voice synthesis with data mined from the web," Proc. the 2020 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020), pp.1979–1989, 2020.

[7] H. Yakura, K. Watanabe, and M. Goto, "Self-supervised contrastive learning for singing voices," IEEE/ACM Trans. on Audio Speech and Language Processing, vol.30, pp.1614–1623, 2022.
[8] B. Sharma, R.K. Das, and H. Li, "On the importance of audio-source separation for singer identification in polyphonic music," Proc. the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019), pp.2020–2024, 2019.
[9] T. Nakatsuka, K. Watanabe, Y. Koyama, M. Hamasaki, M. Goto, and S. Morishima, "Vocal-accompaniment compatibility estimation using self-supervised and joint-embedding techniques," IEEE Access, vol.9, pp.101994–102003, 2021.
[10] Z. Rafii, A. Liutkus, F.R. Stöter, S.I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," IEEE/ACM Trans. on Audio Speech and Language Processing, vol.26, no.8, pp.1307–1335, 2018.
[11] C. Gupta, H. Li, and M. Goto, "Deep learning approaches in topics of singing information processing," IEEE/ACM Trans. on Audio Speech and Language Processing, vol.30, pp.2422–2451, 2022.
[12] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," Proc. the 18th International Society for Music Information Retrieval Conference (ISMIR 2017), pp.745–751, 2017.
[13] 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, no.41, p.1667, 2019.
[14] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation," Proc. the 16th International Workshop on Acoustic Signal Enhancement (IWAENC 2018), pp.106–110, 2018.
[15] A. Jansson, R.M. Bittner, S. Ewert, and T. Weyde, "Joint singing voice separation and F0 estimation with deep U-Net architectures," Proc. the 27th European Signal Processing Conference (EUSIPCO 2019), pp.1–5, 2019.
[16] T. Nakano, K. Yoshii, Y. Wu, R. Nishikimi, K.W.E. Lin, and M. Goto, "Joint singing pitch estimation and voice separation based on a neural harmonic structure renderer," Proc. the 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2019), pp.155–159, 2019.
[17] Y.N. Hung and A. Lerch, "Multitask learning for instrument activation aware music source separation," Proc. the 21st International Society for Music Information Retrieval Conference (ISMIR 2020), pp.748–755, 2020.
[18] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," Proc. the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021), pp.51–55, 2021.
[19] N. Takahashi and Y. Mitsufuji, "D3Net: Densely connected multi-dilated densenet for music source separation," Proc. the 2021 Conference on Computer Vision and Pattern Recognition (CVPR 2021), pp.993–1002, 2021.
[20] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and efficient music source separation tool with pre-trained models," Journal of Open Source Software, vol.5, no.50, p.2154, 2020.
[21] K. Schulze-Forster, C.S.J. Doire, G. Richard, and R. Badeau, "Phoneme level lyrics alignment and text-informed singing voice separation," IEEE/ACM Trans. on Audio Speech and Language Processing, vol.29, pp.2382–2395, 2021.
[22] W. Choi, M. Kim, J. Chung, D. Lee, and S. Jung, "Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation," Proc. the 21st International Society for Music Information Retrieval Conference (ISMIR 2020), pp.192–198, 2020.
[23] W. Choi, M. Kim, J. Chung, and S. Jung, "LaSAFT: Latent source attentive frequency transformation for conditioned source separation," Proc. the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021), pp.171–175, 2021.
[24] Z. Wang, R. Giri, U. Isik, J.M. Valin, and A. Krishnaswamy, "Semi-supervised singing voice separation with noisy self-training," Proc. the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021), 2021.
[25] Q. Kong, Y. Cao, H. Liu, K. Cho, and Y. Wang, "Decoupling magnitude and phase estimation with deep ResUNet for music source separation," Proc. the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021), pp.342–349, 2021.
[26] D. Stöller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," Proc. the 18th International Society for Music Information Retrieval Conference (ISMIR 2017), pp.330–340, 2017.
[27] 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.
[28] T. Nakamura and H. Saruwatari, "Time-domain audio source separation based on wave-u-net combined with discrete wavelet transform," Proc. the 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020), pp.386–390, 2020.
[29] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-learning extractors for music source separation," Proc. the 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020), pp.816–820, 2020.
[30] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," Proc. the 37th International Conference on Machine Learning (ICML 2020), pp.7164–7175, 2020.
[31] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "KUIELab-MDX-Net: A two-stream neural network for music demixing," Proc. Music Demixing Workshop 2021 (MDX 2021), pp.1–7, 2021.
[32] A. Défossez, "Hybrid spectrogram and waveform source separation," Proc. Music Demixing Workshop 2021 (MDX 2021), pp.1–11, 2021.
[33] Y. Hu, Y. Chen, W. Yang, L. He, and H. Huang, "Hierarchic temporal convolutional network with cross-domain encoder for music source separation," IEEE Signal Processing Letters, vol.29, pp.1517–1521, 2022.
[34] L. Prétet, R. Hennequin, J. Royo-Letelier, and A. Vaglio, "Singing voice separation: A study on training data," Proc. the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), pp.506–510, 2019.
[35] A. Cohen-Hadria, A. Roebel, and G. Peeters, "Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation," Proc. the 27th European Signal Processing Conference (EUSIPCO 2019), pp.1–5, 2019.
[36] Y. Wang, D. Stoller, R.M. Bittner, and J.P. Bello, "Few-shot musical source separation," Proc. the 47th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2022), pp.121–125, 2022.
[37] M. Kawamura, T. Nakamura, D. Kitamura, H. Saruwatari, Y. Takahashi, and K. Kondo, "Differentiable digital signal processing mixture model for synthesis parameter extraction from mixture of harmonic sounds," Proc. the 47th IEEE 2022 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2022), pp.941–945, 2022.
[38] Z. Rafii, A. Liutkus, F.R. Stöter, S.I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation." https://doi.org/10.5281/zenodo.1117372.
[39] Z. Rafii, A. Liutkus, F.R. Stöter, S.I. Mimilakis, and R. Bittner, "MUSDB18-HQ - an uncompressed version of MUSDB18." https://doi.org/10.5281/zenodo.3338373.
[40] 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," Proc. the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017), pp.261–265, 2017.
[41] X. Jaureguiberry, E. Vincent, and G. Richard, "Fusion methods for

speech enhancement and audio source separation," IEEE Trans. on Audio Speech and Language Processing, vol.24, no.7, pp.1266–1279, 2016.
[42] J.L. Roux, S. Watanabe, and J.R. Hershey, "Ensemble learning for speech enhancement," Proc. the 23rd IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2013), pp.1–4, 2013.
[43] W. Jiang, S. Liang, L. Dong, H. Yang, W. Liu, and Y. Wang, "Cross-domain cooperative deep stacking network for speech separation," Proc. the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015), pp.5083–5087, 2015.
[44] J. Driedger and M. Müller, "Extracting singing voice from music recordings by cascading audio decomposition techniques," Proc. the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015), pp.126–130, 2015.
[45] E.M. Grais, G. Roma, A.J.R. Simpson, and M.D. Plumbley, "Combining mask estimates for single channel audio source separation using deep neural networks," Proc. the 17th Annual Conference of the International Speech Communication (Interspeech 2016), pp.3339–3343, 2016.
[46] X.L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," IEEE Trans. on Audio, Speech and Language Processing, vol.24, no.5, pp.967–977, 2016.
[47] M. McVicar, R. Santos-Rodriguez, and T.D. Bie, "Learning to separate vocals from polyphonic mixtures via ensemble methods and structured output prediction," Proc. the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016), pp.450–454, 2016.
[48] E. Manilow, P. Seetharaman, F. Pishdadian, and B. Pardo, "Predicting algorithm efficacy for adaptive multi-cue source separation," Proc. IEEE WASPA 2017, pp.274–278, 2017.
[49] A. Défossez, N. Usunier, L. Bottou, and F.R. Bach, "Music source separation in the waveform domain," CoRR, arXiv:1911.13254, pp.1–16, 2021.
[50] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.R. Stöter, A. Défossez, M. Kim, W. Choi, C.Y. Yu, and K.W. Cheuk, "Music demixing challenge 2021," Proc. Music Demixing Workshop 2021 (MDX 2021), pp.1–8, 2021.
[51] I.O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-Mixer: An all-MLP architecture for vision," Proc. the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), pp.24261–24272, 2021.
[52] J.L. Ba, J.R. Kiros, and G.E. Hinton, "Layer normalization," CoRR, arXiv:1607.06450, pp.1–14, 2016.
[53] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," CoRR, arXiv:1606.08415, pp.1–9, 2016.
[54] J. Tae, H. Kim, and Y. Lee, "MLP Singer: Towards rapid parallel korean singing voice synthesis," Proc. the 2021 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2021), pp.1–6, 2021.
[55] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Proc. the 3rd International Conference on Learning Representations (ICLR 2015), pp.1–15, 2015.

Tomoyasu Nakano received the Ph.D. degree in Informatics from University of Tsukuba, Tsukuba, Japan in 2008. He is currently working as the leader of the Media Interaction Group, the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include singing information processing, human-computer interaction, and music information retrieval. He has received several awards including the IPSJ Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ), the Best Paper Award from the Sound and Music Computing Conference 2013, and the Honorable Mention Poster Award from the IEEE Pacific Visualization Symposium 2018. He is a member of the IPSJ and the Acoustical Society of Japan.

Masataka Goto received the Doctor of Engineering degree from Waseda University in 1998. He is currently a Prime Senior Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. Over the past 30 years he has published more than 300 papers in refereed journals and international conferences and has received 58 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth JSPS PRIZE. He has served as a committee member of over 120 scientific societies and conferences, including the General Chair of ISMIR 2009 and 2014. As the research director, he began OngaACCEL project in 2016 and RecMus project in 2021, which are five-year JST-funded research projects (ACCEL and CREST) related to music technologies.