Vol.2022-MUS-135 No.8
2022/9/15

MDX-Mixer: 複数の音楽音源分離モデルによる出力波形を時変混合するシステム

中野 倫靖[1,a)]　後藤 真孝[1,b)]

概要：本稿では、音楽音響信号と、そこから複数の既存の音楽音源分離（MDX）モデルによって分離された音源信号とを、時間変化する重みで混合してより高い分離性能を達成するシステム MDX-Mixer を提案する。従来、歌声分離や音楽音源分離において、時不変もしくは時変な正の混合重みによって、分離信号を混合する研究があった。これらに対して本研究では、分離された音源信号に加えて、分離前の音楽音響信号も活用した時変な混合を行う。時変重みは、音楽音響信号及びその分離信号を短いセグメントに分割してモデル化して推定し、音楽音響信号を考慮するために負の重みも許容する。本稿では、1x1 convolution を用いて時不変重みを推定する比較モデルと、コンピュータービジョン分野で提案されている MLP-Mixer レイヤーをセグメント毎に適用して時変重みを推定する MDX-Mixer の二種類を新たに提案し、MUSDB18-HQ データセットを用いて、音源対歪み比 (SDR) に基づいて評価する。その結果、MDX-Mixer は、混合に用いた最先端の MDX モデルよりも高い SDR を達成することを示す。

1. はじめに

音楽音源分離（music demixing: MDX、もしくは music source separation: MSS）の課題は、実世界の音楽音響信号からボーカル、ドラム、ベース等の個々の音源信号を分離することである。高性能な音楽音源分離は、個々の音源の特性を分析および利用する様々なアプリケーションにとって不可欠な技術である。実際、音楽鑑賞のための個々の楽器へのエフェクト追加 [1] や音量の調整 [1–4]、好みのパートの音量調整による人工内耳ユーザーの音楽体験向上 [5]、歌声合成 [6]、歌声の特徴表現獲得 [7]、歌手同定 [8]、歌声と伴奏の相性推定 [9]、等に利用されている。また、音楽制作に関する商用ソフトウェアへの導入事例がある[*1]。

MDX では、汎化に優れた高性能な音源分離フレームワーク構築を目的として、モデルアーキテクチャと学習方法の開発、多様で膨大な学習データの準備・拡張に関して、研究が取り組まれてきた。モデルアーキテクチャとしては、深層ニューラルネットワーク（Deep Neural Network: DNN）が最も高性能なフレームワークの 1 つとして広く使用されており [16, 17]、現在の深層 MDX モデルは以下の 4 種類

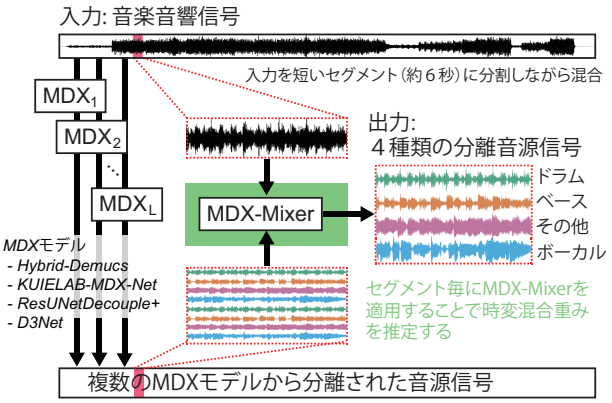

図 1　MDX-Mixer の概要。音楽音響信号と、そこから複数の既存の音楽音源分離 (MDX) モデルによって分離された音源信号の全てを約 6 秒毎に混合して最終的な分離信号を得る。

に大別される:

(1) 振幅スペクトル領域における分離 [10, 18–26]
(2) 複素スペクトル領域における分離 [13, 27–29]
(3) 波形領域における分離 [11, 30–34]
(4) 波形と複素スペクトルのハイブリッド分離 [14, 15]

その他、学習データの量・多様性の増加 [29, 35, 36] 、Few-shot 学習による少数学習データへの対処 [37]、混合音から楽音合成用のパラメータ推定 [38] などが研究されている。

MDX の研究の多くでは、ドラム、ベース、ボーカル、それ以外の 4 音源を対象とした分離手法の音源対歪み比 (source-to-distortion ratios: SDRs) による評価がなされて比較される。**表 1** に、2022 年 7 月の時点で、音源ごとに音源対歪

[1] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[a)] t.nakano@aist.go.jp
[b)] m.goto@aist.go.jp

[*1] 例えば、Audionamix XTRAX STEMS (`https://audionamix.com/xtrax-stems/`)

表 1 音楽音源分離における SOTA モデルの MUSDB18(-HQ) における論文記載の SDRs。“All” は、4 音源の結果の平均を意味する。**太字**は各データセットにおける最大値、下線は MUSDB18 における 2 番目に高い値を示す。‘*’ が併記されたモデルは MUSDB18-HQ で評価された。† の併記モデルは全 50 曲における SDR の中央値によって評価され、それ以外は “median of frames, median of tracks” によって評価された。

	Test SDR in dB				
Model	All	Drums	Bass	Other	Vocals
D3Net [10]	6.01	7.01	5.25	4.53	7.24
Demucs [11]	6.28	6.86	7.01	4.42	6.84
CDE-HTCN [12]	**6.89**	**7.33**	**7.92**	4.92	7.37
ResUNetDecouple+† [13]	6.73	6.62	6.04	**5.29**	**8.98**
KUIELAB-MDX-Net*† [14]	7.54	7.33	7.86	**5.95**	**9.00**
Hybrid-Demucs* [15]	**7.68**	**8.24**	**8.76**	5.59	8.13

み比 (SDR) が最も高い 6 つの state-of-the-art (SOTA) モデルの SDRs を示す。上の 4 行は MUSDB18 [39] の SOTA モデルを示し、下の 2 行は MUSDB18-HQ [40] (MUSDB18 の周波数帯域幅拡張バージョン) の SOTA モデルを示す。

表 1 より、各音源における SDR が最高となる分離波形を出力するモデルがそれぞれ異なる場合があることが分かる。ここから、「複数の MDX モデルによる分離音源信号を混合することにより、すべての音源に対して最高性能を得ることができるか、またはそれを超えることができるか？」という問いが考えられる。実際、分離性能を向上させるために、分離された音源信号を時不変に混合する 2 つの MDX 研究がある [14, 41]。これらは、2 種類のモデルによって分離された同一種類の音源同士を正の重みを用いて混合していた。また、最適な重みは音楽の内容に応じて時々刻々と変化する可能性があることから、歌声分離信号のみを対象として、時変な正の混合重みを推定する手法が提案され、その有効性が示された [42]。

これらに対して本研究では、分離された信号に加えて、分離前の音楽音響信号（以降、単に音楽音響信号と呼ぶ）を活用し、また異なる種類の音源も含めて時変な混合を行う MDX-Mixer を提案する (**図 1**)。時変重みは、図 1 のように、音楽音響信号及びその（SOTA MDX モデルによる）分離信号を短いセグメントに分割して混合重みを推定することで実現できる。ここで、音楽音響信号と他の種類の分離音源を活用するために、負の重みも許容する。負の重みは、異なる種類の音源の残留信号の除去や、対象とする音源信号の強調に対して、有効である可能性がある。

2. 関連研究

従来、複数の音源分離モデルもしくはその推定結果を混合する Blending, Fusion, Ensemble, Combine 等と呼ばれる研究がなされてきた [14, 41–49]. 音声強調や音声分離などにおいては、振幅スペクトルもしくはそのマスクを対象に、モデルの統合や推定マスクの混合がなされた [43, 44, 46, 47]。

音楽音源分離もしくは歌声・伴奏分離においては、まず、複数の分離手法の結果を提案モデルの入力として活用する研究がある。歌声（伴奏音）分離を対象として、McVicar *et al.* [48] は、複数の音源分離手法による出力を特徴ベクトルとして用い、条件付き確率場によって振幅スペクトルマスクを推定する手法を提案した。

次に、複数の分離手法を提案モデルの構成要素の一つとして用いる研究がある。Driedger *et al.* [45] は、振幅スペクトルから調波構造など個別の特徴に基づいた分離手法を多段に適用する方法を提案した。

そして最後に、本研究と特に関係する、音楽音源分離における分離波形を選択 [49] もしくは混合 [14, 41, 42] する研究がある。Manilow *et al.* [49] は、複数の分離手法の SDR を短時間毎に推定する DNN モデルを学習し、予測 SDR を最大化するように分離結果を選択する手法を提案した。Uhlich *et al.* [41] 及び Kim *et al.* [14] は、時不変な正の混合重みによって分離された音源信号を混合した。2 つの MDX モデル（model1 と model2）で分離された音源信号を $x_{i,\mathrm{model1}}(t)$ と $x_{i,\mathrm{model2}}(t)$ とすると、音源 i 毎の時不変重み w_i を用いて以下のように混合される。

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

Uhlich *et al.* [41] は、DSD100 Dev セットの平均 SDR を最大化する、全音源に共通の時不変な重み w_i を決定した。最適な $w_i = 0.25$ を使用して、Feed-forward モデル (model1) と BLSTM モデル (model2) によって分離された信号を混合した。Kim *et al.* [14] は音源依存の重み w_i を使用して、修正された TFC-TDF-U-Net [27] (model1) と Demucs [11] (model2) によって分離信号を混合した。具体的には、w_i は、MDX Challenge 2021 [50] において、ベース、ドラム、その他、およびボーカルに対して 0.5、0.5、0.7、および 0.9 としていた[*2]。

また、分離信号の時変な混合に関しては、歌声分離を対象として、Jaureguiberry *et al.* [42] による提案がある。これは、音楽音響信号と複数の分離歌声信号の短時間のパ

[*2] https://github.com/kuielab/mdx-net/blob/Leaderboard_A/README_SUBMISSION.md

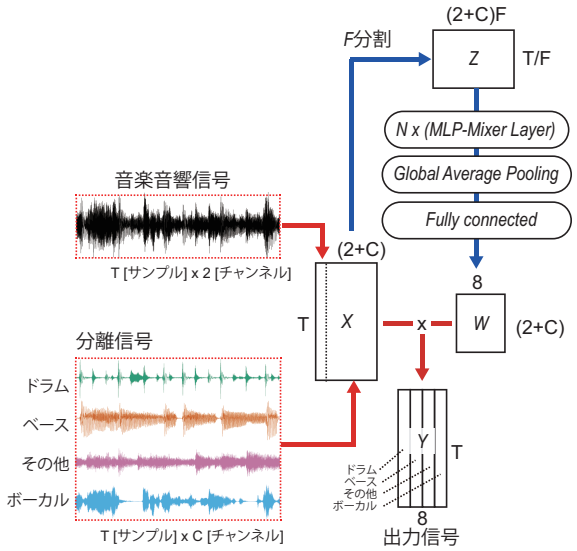

図 2 MDX-Mixer のシステム概要。時変混合重み $\mathbf{W}$ を推定する。青矢印で示された処理が実施されない場合は 1x1 convolution によって時不変な重みが推定され、その場合 $\mathbf{W}_0$ と呼ぶ。

ワー スペクトルを入力として、時刻 n で $\forall w_{i,n} \geq 0$ を条件とする正の混合重み $\sum_i w_{i,n} = 1$ を推定する手法である。

しかしこれまで、波形領域における分離信号の混合について、時変混合重みの有効性が歌声以外の分離で評価されたことはなく、また、負の混合重みを対象として音楽音響信号や他の種類の音源が活用されたことはなかった。

3. 手法

本稿では音楽音響信号と、分離後の音源信号（以降、単に分離信号と呼ぶ）を波形領域で混合するために、(1) 1x1 convolution を用いて時不変重みを学習する比較システムと、(2) 短く分割されたセグメント毎（約 6 秒）に MLP-Mixer レイヤー [51] を用いて時変重みを学習する MDX-Mixer の二種類を新たに提案する。

ここで、これらの提案システムへ入力する信号は、ステレオの音楽音響信号[*3]及び複数の MDX モデルによって得られる複数の分離信号をチャンネル方向に重ね合わせた複数チャネル信号であり、$\mathbf{X} \in \mathbb{R}^{T\times(2+C)}$ とする。$\mathbf{X}$ は、T サンプルの 2 チャネル音楽音響信号の行列と T サンプルの C チャネル分離信号の行列を連結することによって得る。ここで、三つの MDX モデルからそれぞれ 4 音源を分離した場合、$C = 12$ となる。各 MDX モデルは、必ずしも 4 つの対象音源すべてを出力する必要はなく、例えばボーカルのみの出力でも良い。

最終的な出力としてのステレオ分離信号 $\mathbf{Y}$ は、ドラム、ベース、ボーカル、それ以外の 4 音源であり、8ch となる。

3.1 1x1 convolution に基づく時不変混合システム

Kim *et al.* [14] は、音楽音響信号と「単一の」MDX モデルによる分離信号に対して、他の音源の残留信号の除去を目的として 1x1 convolution を用いた。それに対して本稿では、音楽音響信号と「複数の」MDX モデルによる分離信号を混合するために、1x1 convolution を使用する。

1x1 convolution は、**図 2** の青矢印の処理を行わない MDX-Mixer の特殊なケースであり、推定する重みは時不変であることから $\mathbf{W}_0$ と呼ぶ。4 つの音源の 8 ch 分離信号 $\mathbf{Y}$ を得るために、$\mathbf{X}$ との内積 $\mathbf{Y} = \mathbf{X}\mathbf{W}_0$ をとる時不変な重み行列 $\mathbf{W}_0 \in \mathbb{R}^{(2+C)\times 8}$ を推定する処理は、1x1 convolution を適用することに相当する。$\mathbf{W}_0$ は入力信号やその時刻によらず同じ重みとなることから、各 MDX モデルの対象音源毎への影響の度合いを表現する。

3.2 MDX-Mixer: 音楽音響信号と複数分離波形の時変混合システム

図 2 に、MDX-Mixer の概要を示す。時変混合のためには、入力信号のコンテキストに適した重みを推定する必要があることから、1x1 convolution のようなチャネル間の関係 （音源全体のコンテキスト）だけでなく、チャネル内の関係 （各チャネル内の時間コンテキスト）も併せてモデル化することが望ましい。1x1 convolution はチャネル間の全結合で表現できるが、それに加えてチャネル内の全結合も考慮する MLP-Mixer レイヤー [51] を利用する。MLP-Mixer レイヤーは、コンピュータービジョン分野で提案され、シンプルな構造、高性能、低学習コスト、および高い推論スループットの利点がある。この MLP-Mixer レイヤーを通して、時変混合重み $\mathbf{W} \in \mathbb{R}^{(2+C)\times 8}$ を推定し、$\mathbf{X}$ と $\mathbf{W}$ (*i.e.*$\mathbf{Y} = \mathbf{X}\mathbf{W}$) の積として 4 つのステレオ音源の 8ch 出力信号 $\mathbf{Y}$ を得る。

MLP-Mixer レイヤーの構造を**図 3** に示す。文献 [51] では、入力画像をパッチ分割して複数チャネルの信号として用い、個々の分割画像（チャネル）内の全結合である *token-mixing MLP* と、分割画像（チャネル）間の全結合である *channel-mixing MLP* を繰り返すことで画像クラスを推定していたが、本稿では、それを音響信号に用いる。そのまま適用すると、T サンプル、$(2+C)$ch の行列を入力とすることになるが、T が大きいと token-mixing MLP に必要な重み行列 $\mathbf{W}_{\text{token}}$ のサイズが巨大になる。そこで、そのサイズを減らすために、T サンプルを F 分割してチャネル方向に連結し、行列 $\mathbf{Z} \in \mathbb{R}^{T/F\times(2+C)F}$ を得る。

現在の実装では、$T = 2^{18}$（サンプリング周波数 44.1 kHz で約 6 秒）を用いる。ch 数を仮に $(2+C) = 12$ とすると、分割しない場合の token-mixing MLP の重み行列 W_{token} のサイズは $(2^{18})^2$ であり、channel-mixing MLP の重み行列 W_{channel} のサイズは 12^2 となる。これらを $F = 2^8$ で分割すると、サイズがそれぞれ $(2^{10})^2$ と $(12 \times (2^8))^2$ で 0.000153 倍となり減少する。このような分割により、token-mixing MLP は分割されたセグメント内の関係をモ

[*3] 各信号の左（L）と右（R）それぞれを 1ch として扱う。

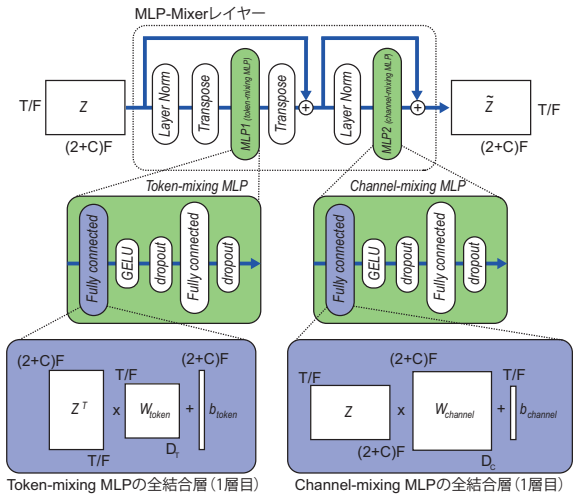

図 3 MLP-Mixer レイヤーの概要。

デル化でき、channel-mixing MLP はそれらのセグメント間の関係をモデル化できることになる。

図 2 に示すように、分割行列 $\mathbf{Z}$ は、N 層の MLP-Mixer レイヤーを通る。各 MLP-Mixer レイヤーは、Layer Normalization [52]、スキップ接続、Gaussian Error Linear Unit (GELU) [53]、ドロップアウトを含む token-mixing MLP と channel-mixing MLP で構成される（図 3）。token-mixing MLP と channel-mixing MLP には、それぞれ 2 つの全結合層があるため、時間とチャネルに対してそれぞれ D_T と D_C と呼ばれる層間の隠れ次元の数を設計できる。MLP-Mixer レイヤーの後には、チャネル全体で平均化して、データを要約しながら要素数を減らす Global Average Pooling が続き、最後に全結合層を介して重み行列 $\mathbf{W}$ を得る。文献 [51] では、分割画像は Embedding レイヤーを経由して与えられていたが、今回は用いなかった[*4]。

このようにして推定された時変な $\mathbf{W}$ はコンテキストを考慮し、その重みは音楽音響信号のセグメントの内容に従って変化する。つまり $\mathbf{W}$ は、$\mathbf{W}_0$ と異なり、既存の各 MDX モデルの対象音源毎への影響の度合いと、得意な楽曲も表現することが期待される。

4. 実験

提案した 1x1 convolution に基づくシステムと MDX-Mixer の有効性を評価するために、MDX 研究で標準的に用いられる MUSDB18-HQ データセット [40] を使用して、各種パラメータを学習した。86 曲を学習データ、14 曲をバリデーションデータ、50 曲をテストデータとした。

「ドラム」、「ベース」、「その他」、「ボーカル」の 4 つの音源を分離の対象とし、音楽音響信号はサンプリング周波数 44.1 kHz のステレオ音源とした。

[*4] 入力信号を直接重み付けて混合することから、埋め込みを経由しない方が適切な表現ではないかと考えたため。ただし、埋め込みを含めて、音楽音響信号により適した何らかの変換が行える可能性はあるので、その検討は今後の課題である。

分離性能は、*museval* python package[*5] を使用して SDR を計算して評価した。従来研究（[10,15] 等）と同様に、各曲 1 秒毎に中央値を計算してその中央値で各曲の SDR として、全曲すべての中央値を各音源の SDR とした（*i.e.*“median of frames, median of tracks”）。

4.1 利用する既存 MDX モデル

混合に用いる分離信号を得るための既存 MDX モデルとして、研究目的で利用可能な以下の 4 つの事前学習済みモデルを用いた。これらは全て、Web 上で公開されている。

モデル A: “d3net-mss”: D3Net [10] の事前学習済み公開モデル[*6]. 表 1 において、MUSDB18 を対象とした際にドラムの SDR が 2 番目に高い。今回用いる他の 3 モデルと異なり、位相を明示的に推定しない。

モデル B: “ResUNet143 Subband vocals”: ResUNetDecouple+ [13] の事前学習済み公開モデル[*7]。MUSDB18 においてボーカルとその他の SDR が最も高く、特にボーカルは他より 1.6 dB 以上高い（表 1）。ボーカル・伴奏分離モデルのみが公開されているため、ボーカル分離信号のみを使用した。

モデル C: “kuielab_mdxnet_A”: KUEILAB-MDX-Net [14] の事前学習済み公開モデル[*8]。MUSDB18-HQ におけるボーカルおよびその他の SDR が最も高い（表 1）。最終層に、1x1 convolution で音楽音響信号と 4 音源を混合する処理が含まれる。

モデル D: “mdx” Hybrid-Demucs [11] の事前学習済み公開モデル[*9]。MUSDB18-HQ におけるドラム、ベースの SDR が最も高く、また 4 音源の平均（“All”）も最も高い（表 1）。

これらは、MUSDB18-HQ または MUSDB18 の学習用データのみで（テストデータを使用せずに）学習された。

これらのモデルの MUSDB18-HQ における評価結果を**表 2** に示す。学習データセット、モデル、評価方法等が本稿と元論文では異なることが原因で、表 1 とは完全には一致しないことに注意する。これらを混合することで、これらの SDR を超える性能を得ることが目的である。

4.2 1x1 convolution 及び MDX-Mixer の学習

提案システムは、音楽音響信号のみを用いるか、もしくは、それに加えて 4.1 に示されている MDX モデルを 1 つ以上使用して得た分離信号を用いて学習した。

時間方向を F 分割するため、サンプル数 T は 2 の累乗

[*5] https://github.com/sigsep/sigsep-mus-eval
[*6] https://github.com/sony/ai-research-code/tree/master/d3net/music-source-separation
[*7] https://github.com/bytedance/
[*8] https://github.com/kuielab/mdx-net-submission/tree/leaderboard_A
[*9] https://github.com/facebookresearch/demucs

表 2 事前学習済み MDX モデルの MUSDB18-HQ（テスト用データ）における SDR。“*” が付与されたモデルは MUSDB18-HQ（学習用データ）を用いて学習された。各音源における最高値を**太字**で示す。

Model		Test SDR in dB				
ID	Name	All	Drums	Bass	Other	Vocals
A	“d3net-mss” (D3Net [10])	5.93	6.59	5.25	4.82	7.08
B	“ResUNet143 Subband vocals” (ResUNetDecouple+ [13])	N/A	N/A	N/A	N/A	8.21
C	“kuielab_mdxnet_A” (KUIELAB-MDX-Net [14])*	7.47	7.20	7.83	**5.90**	**8.97**
D	“mdx” (Hybrid-Demucs [15])*	**7.77**	**8.21**	**9.28**	5.50	8.10

表 3 1x1 convolution に基づく時不変混合重み $\mathbf{W}_0$ を推定するシステムの MUSDB18-HQ における SDR。表 2 の最高値を超えた場合は**太字**で表記し、特に最高値を下線で示す。“*” の表記は、その音源の分離信号が混合されなかったことを意味する。例えば ID:1-0 はいずれの既存 MDX モデルも使用しておらず、音楽音響信号のみを入力としたことから、全ての音源で分離信号を混合に用いていない。ID: 1-5, 1-6, 1-7 については、同一条件でランダムシードを変えた 3 回の実行結果を示した。

1x1 convolution					Test SDR in dB				
ID	A	B	C	D	All	Drums	Bass	Other	Vocals
1-0					0.68	0.36*	0.85*	1.32*	0.20*
1-1	✓				5.94	6.59	5.25	4.85	7.05
1-2		✓			2.82	0.58*	1.13*	1.48*	8.09
1-3			✓		7.36	7.19	7.46	5.98	8.8
1-4				✓	7.73	**8.25**	9.00	5.55	8.11
1-5-1		✓		✓	**7.83**	**8.29**	9.00	5.70	8.31
1-5-2		✓		✓	**7.81**	**8.24**	9.00	5.71	8.30
1-5-3		✓		✓	**7.83**	**8.25**	8.98	5.71	8.37
1-6-1			✓	✓	**8.13**	**8.34**	9.04	**6.18**	8.97
1-6-2			✓	✓	**8.00**	**8.37**	8.87	**6.18**	8.59
1-6-3			✓	✓	**8.04**	**8.39**	9.04	**6.17**	8.55
1-7-1		✓	✓	✓	**8.03**	**8.26**	9.06	**6.23**	8.57
1-7-2		✓	✓	✓	**8.05**	**8.35**	8.99	6.31	8.54
1-7-3		✓	✓	✓	**8.16**	**8.34**	9.08	**6.19**	**9.05**

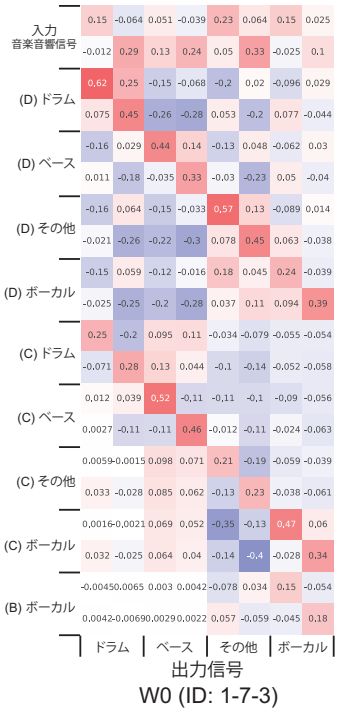

図 4 時不変混合重み $\mathbf{W}_0$ の例。

に設定した。本稿では 1x1 convolution と MDX-Mixer のいずれも、信号を シフト間隔を 2^{15}（約 0.7 秒）とした $T = 2^{18}$（約 6 秒）のセグメントを学習に用いた。T は、波形ベースの MDX 研究で使用された長さ（*i.e.*10 秒 [11]）を参考にし、また、MDX モデル B と D を用いた条件（表 3 の ID:1-5-1 等）での予備実験として、T が 2^{18} と 2^{19} のいずれでも SDR が同程度であったため、必要メモリが小さく、より高い時間分解能で推定できる $T = 2^{18}$ を選択した。

MDX-Mixer 特有のハイパーパラメータとしては、$F = 2^7$、MLP-Mixer レイヤー数 $N = 8, 16$、およびドロップアウト確率 p を 0 または 0.2 とした。N と p は、MLP-Mixer レイヤーを使用した研究 [51, 54] を参考に決定した。隠れ層 D_T と D_C の次元数は、それぞれ $\mathbf{Z}$ と同サイズ、つまり $D_T = T/F$, $D_C = (2 + C)F$ とした。

1x1 convolution は、Adam [55] を用いて、学習率 0.0003 で 50 エポック学習した。また、MDX-Mixer は、同じ条件下で 100 エポック学習した。1x1 convolution に基づくシステムにおいては、20 エポック前後でバリデーションロスが収束したが、MDX-Mixer では、100 エポックの間にロスが変動し、徐々に減少する傾向があった。どちらのシステムも、バッチサイズ 64 として、正解分離信号からの L1 ロスに基づいて学習した。波形は、音楽音響信号の振幅平均が 0、標準偏差が 1 となるように正規化した。

各学習条件に対して、バリデーションロス最小のモデルを評価に用いた。テストデータの分離では、音楽音響信号とその MDX モデルによる分離信号は、シフト幅 $T/4$ で固定長 T のセグメントに分割され、それらの混合結果を重み付きオーバーラップ加算して最終信号 $\mathbf{Y}$ を得た。

4.3 実験結果

表 3 と**表 4** に、異なるハイパーパラメータによって学習されたシステムの結果を示す。列 “A”、“B”、“C”、および “D” のチェックマークは、分離信号を得るために使用される MDX モデルを示しており、いずれもチェックされていない場合は、音楽音響信号のみを混合したことを意味する。ここで、モデル B はボーカル分離信号のみを出力し、それ

表 4 時変混合重み $\mathbf{W}$ を推定する MDX-Mixer の MUSDB18-HQ における SDR。表 2 の最高値を超えた場合は**太字**で表記し、特に最高値を下線で示す。“*” の表記は、その音源の分離信号が与えられなかったことを意味する。

MDX-Mixer							Test SDR in dB				
ID	A	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	5.93	6.58	5.20	4.81	7.15
2-2		✓			8	0	2.90	1.13*	0.97*	1.43*	8.06
2-3			✓		8	0	7.45	7.19	7.83	5.87	8.92
2-4				✓	8	0	7.72	**8.25**	8.98	5.54	8.09
2-5		✓		✓	8	0	**8.04**	**8.65**	9.17	5.77	8.59
2-6			✓	✓	8	0	**8.16**	**8.24**	9.22	**6.19**	8.97
2-7		✓	✓	✓	8	0	**8.21**	**8.29**	9.19	**6.26**	**9.08**
2-8		✓		✓	8	0.2	**7.94**	**8.42**	9.18	5.76	8.42
2-9			✓	✓	8	0.2	**8.16**	**8.24**	9.28	**6.27**	8.85
2-10		✓	✓	✓	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	5.97	6.56	5.28	4.84	7.20
3-2		✓			16	0	2.84	1.03*	0.85*	1.41*	8.10
3-3			✓		16	0	7.45	7.19	7.78	**5.93**	8.91
3-4				✓	16	0	7.72	**8.26**	8.97	5.54	8.11
3-5		✓		✓	16	0	**7.96**	**8.40**	9.04	5.80	8.57
3-6			✓	✓	16	0	**8.16**	**8.29**	9.16	**6.16**	**9.02**
3-7		✓	✓	✓	16	0	**8.21**	**8.54**	9.10	**6.17**	**9.01**
3-8		✓		✓	16	0.2	**7.95**	**8.63**	9.12	5.73	8.34
3-9			✓	✓	16	0.2	**8.21**	**8.29**	9.26	**6.25**	**9.03**
3-10		✓	✓	✓	16	0.2	**8.15**	**8.27**	9.20	**6.23**	8.90

表 5 MUSDB18-HQ を対象とした既存 MDX モデルによる SDR の最高値（表 2）、1x1 convolution に基づくシステムの 3 回の実行結果（表 3）の平均値、及び MDX-Mixer による異なるレイヤー数 N とドロップアウト p を用いた結果（表 4）の平均値。例えば “CD” は、音楽音響信号とモデル C と D による分離信号を混合したことを意味し、MDX-Mixer の場合は ID: 2-6, 2-9, 3-6, 3-9 である。

ID	All	Drums	Bass	Other	Vocals
max(表 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

表 6 音楽音響信号を混合に用いない場合の MDX-Mixer の MUSDB18-HQ における SDR の 4 条件の平均。音楽音響信号を混合に用いた表 5 の同一条件の値を超えた場合は**太字**とし、下がった場合は ↓ を併記する。

ID	All	Drums	Bass	Other	Vocals
MDX-Mixer（音楽音響信号を使用しない）					
mean(CD)	8.17	8.27	9.23	6.14↓	**9.05**
mean(BCD)	8.15↓	8.27↓	9.13↓	6.15↓	**9.06**

表 7 T と F を変更した場合の MDX-Mixer の MUSDB18-HQ における SDR の 4 条件の平均。表 5 の同一条件の値を超えた場合は**太字**とし、下がった場合は ↓ を併記する。

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**

以外のモデルは 4 つの音源分離信号すべてを出力した。モデル A による分離信号は、個々の音源の位相が明示的に推定されない MDX モデルにおける提案システムの有効性を議論するために単独で使用した。

表 5 に評価結果の統計値として、1x1 convolution に基づいたシステムにおいて 3 回の実行結果（同じハイパーパラメータと異なるランダムシードを使用した学習）の平均と、 MDX-Mixer において同じ MDX モデルを用いた異なるハイパーパラメータの結果の平均を示す。

システムの有効性をより詳細に検証するために、モデル C と D を用いる条件（ID: 2-6, 2-9, 3-6, 3-9）と、モデル B、C、および D を用いる条件（ID: 2-7, 2-10, 3-7, 3-10）において、混合に音楽音響信号を用いなかった場合の SDR 平

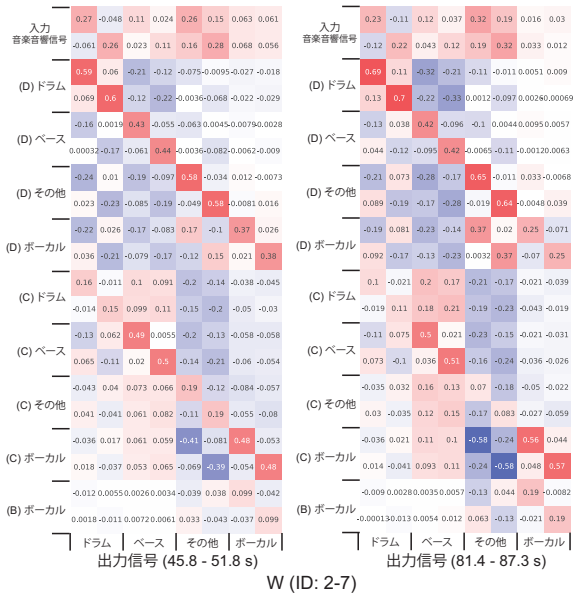

図 5 MDX-Mixer により推定された、異なる時刻における時変混合重み $\mathbf{W}$ の例（楽曲名: “The Doppler Shift - Atrophy”）

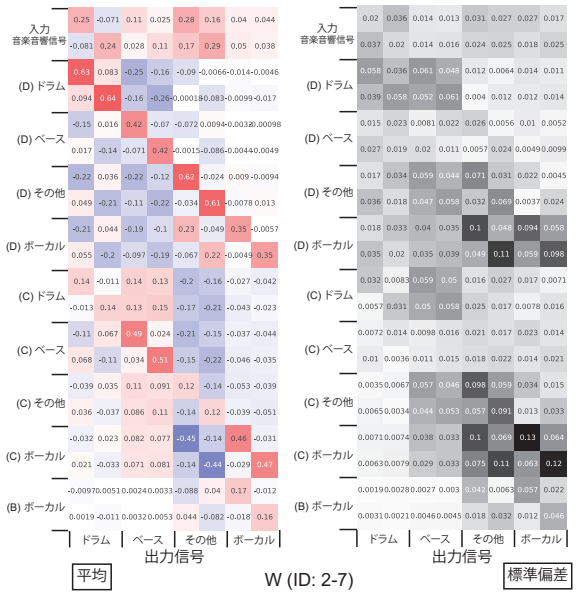

図 6 MDX-Mixer により推定された $\mathbf{W}$ の 1 曲中の平均と標準偏差の例（楽曲名: “The Doppler Shift - Atrophy”）

均を**表 6** に示す。また、セグメント長 T と分割係数 F を変更した場合の結果について、同様に SDR 平均を**表 7** に示す。ここで、一つのトークンサイズ T/F（token-mixing MLP のサイズ）が一定となるように $(T = 2^{17}, F = 2^6)$ と $(T = 2^{16}, F = 2^5)$ とした。

最後に、時不変混合重み $\mathbf{W}_0$ と時変混合重み $\mathbf{W}$ の推定例を、それぞれ**図 4** と**図 5** に示す。また、$\mathbf{W}$ の 1 曲中の平均と標準偏差を**図 6** に示す。$\mathbf{W}_0$ は ID:1-7-3、$\mathbf{W}$ は ID:2-7 によって、MUSDB18-HQ テスト用データ “The Doppler Shift - Atrophy” に対する推定結果である。列が混合に用いた音源であり、B,C,D は MDX モデルの ID である。

4.4 考察

まず、単一の MDX モデルに 1x1 convolution と MDX-Mixer を適用した結果（ID: 1-/2-/3-1,2,3,4）、これらの SDR は表 2 と比較して改善されたとは言えない。しかし、分離信号を用いなかった条件（ID：1-0、2-0、3-0）と、モデル B によるボーカル分離信号のみを用いた条件（ID：1-2、2-2、3-2）の結果から、ボーカル以外の音源に対する SDR が向上した。このことから、Kim *et al.* [14] の報告にあるように、他の音源を用いて分離性能を改善できる可能性があることが部分的に示唆された。

次に、1x1 convolution を複数の MDX モデルに適用した結果（表 3: ID: 1-5, 1-6, 1-7 においてそれぞれ 3 回実行）は、表 5 に示すように、単一 MDX モデルを用いるよりも平均的に高い性能を得ることが “All” 列から分かる。“All” 列以外の、各音源に対する MDX モデルの SDR 最大値（表 2）を平均すると 8.09 dB であることから、音源毎に最大値を取る MDX モデルを手動で選択した場合と同程度であると言えるが、それを自動的に決定できる利点がある。

そして、MDX-Mixer を複数の MDX モデルに適用した場合（表 4: ID: 2-5 から 2-10、および 3-5 から 3-10）、今回の条件では、MLP-Mixer レイヤーの数 N とドロップアウト確率 p の違いが、SDR にほとんど影響を与えないことが示唆されたが、1x1 convolution に基づくシステムと比較して、SDR は平均して改善した（表 5）。さらに “All” 列は、各音源に対する MDX モデルの SDR 最大値（表 2）を平均した 8.09 dB よりも高い値を取ることから、音源毎に最大値を取る MDX モデルを手動で選択するよりも性能を上げられる可能性がある。モデル C と D を用いた場合（ID: 3-9）、またはモデル B、C、および D を使用した場合（ID: 2-7、3-7） に最も改善され、“All” は 8.21 dB となった。これは、SOTA の MDX モデルの最大値 7.77 dB から 0.44 dB 以上の SDR の改善となる。以上から、時変混合重みを推定する MDX-mixer の有効性が示された。

図 4〜図 6 からは、各 MDX モデルによる SDR が高い（得意な）音源の重みが大きかった。図 5 と図 6 からは、重みが時間とともに変化しており、標準偏差からは、この例では、特にボーカルとその他に関して変動が大きいことが分かる。また、分離信号のみでなく、音楽音響信号に対しても、比較的大きな正の重みが推定されていることから、音楽音響信号が活用された可能性を示唆している。仮に、音楽音響信号を混合に用いなかった場合（表 6）、ドラム、ベース、その他の SDR が平均的には減少し、逆にボーカルは音楽音響信号を含めない場合の平均 SDR が高かった。図 6 の平均値からは、音楽音響信号のボーカルに対する重みが相対的に小さく、それ以外は相対的に大きかったことから、音楽音響信号はボーカル以外 3 種の音源に対して影響力を持っている可能性がある。さらに、音響信号及び分離信号において、それぞれ負の重みも推定されており、他

の音源に残留した音を減衰する等の効果が期待される。

また、セグメント長 T と分割係数 F を変更しても、表 7 から、平均的には性能が変わらなかった。ただし、ドラムとボーカルに向上が、ベースに減少が見られたことから、これらを調整して結果をチューニングできる可能性はある。

最後に、MDX モデル B、C、および D は、波形または位相スペクトルを使用するため、波形領域での混合に特に適している可能性がある。一方、位相を推定しないモデル A も性能の低下は見られなかったため、提案システムは様々なモデルに利用できる可能性がある。

5. おわりに

本稿では、複数の SOTA の MDX モデルによる分離音源信号を活用するシステム MDX-Mixer を提案し、個々の MDX モデルの得意な音源とコンテキスト（楽曲）に対処できることを示した。本稿の貢献は以下の通りである。

- 複数の既存 MDX モデルを使用して分離された音源信号を混合する 1x1 convolution を提案した。
- 時変の混合重みを推定する MDX-Mixer を提案した。
- 既存 MDX モデルを使用して、1x1 convolution に基づくシステムと MDX-Mixer の両方の SDR を向上させることができることを示した。結果から、時変重みを推定する MDX-Mixer がより優れており、音源毎に最大値を取る既存 MDX モデルを手動で選択するよりも性能を上げられる可能性があることを示した。
- 図 4〜図 6 から、音楽音響信号が正の重みで混合されて活用される可能性を示した。また、音楽音響信号と分離信号において負の重みも推定されていて、他音源の残留信号の除去等に使われた可能性がある。

本稿では、コンテキストをモデル化するために、まずはシンプルな構造を持つ MLP-Mixer レイヤーを導入したが、今後の課題として、他のコンテキストモデリング手法（*e.g.*Transformer、CNN 等）の検討がありうる。さらに、より効果的なハイパーパラメータの選択と MDX-Mixer の学習フレームワークの改善に取り組む必要がある。

謝辞 本研究の一部は JST CREST JPMJCR20D4 と JSPS 科研費 JP21H04917 の支援を受けた。

参考文献

[1] Woodruff, J., Pardo, B. and Dannenberg, R.: Remixing Stereo Music with Score-Informed Source Separation, *Proc. ISMIR 2006* (2006).
[2] Gillet, O. and Richard, G.: Extraction And Remixing Of Drum Tracks From Polyphonic Music Signals, *Proc. IEEE WASPAA 2005*, pp. 315–318 (2005).
[3] Yoshii, K., Goto, M. and Okuno, H. G.: INTER:D: A Drum Sound Equalizer for Controlling Volume and Timbre of Drums, *Proc. EWIMT 2005*, pp. 205–212 (2005).
[4] Itoyama, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H. G.: Instrument Equalizer for Query-by-Example Retrieval: Improving Sound Source Separation Based on Integrated Harmonic and Inharmonic Models, *Proc. ISMIR 2008* (2008).
[5] Pons, J., Janer, J., Rode, T. and Nogueira, W.: 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] Ren, Y., Tan, X., Qin, T., Luan, J., Zhao, Z. and Liu, T.-Y.: DeepSinger: Singing Voice Synthesis with Data Mined From the Web, *Proc. KDD 2020*, pp. 1979–1989 (2020).
[7] Yakura, H., Watanabe, K. and Goto, M.: Self-Supervised Contrastive Learning for Singing Voices, *IEEE/ACM TASLP*, Vol. 30, pp. 1614–1623 (2022).
[8] Sharma, B., Das, R. K. and Li, H.: On the Importance of Audio-source Separation for Singer Identification in Polyphonic Music, *Proc. Interspeech 2019*, pp. 2020–2024 (2019).
[9] Nakatsuka, T., Watanabe, K., Koyama, Y., Hamasaki, M., Goto, M. and Morishima, S.: Vocal-Accompaniment Compatibility Estimation Using Self-Supervised and Joint-Embedding Techniques, *IEEE Access*, Vol. 9, pp. 101994–102003 (2021).
[10] Takahashi, N. and Mitsufuji, Y.: D3Net: Densely connected multidilated DenseNet for music source separation, *CoRR*, Vol. abs/2010.01733 (online), available from ⟨https://arxiv.org/abs/2010.01733⟩ (2020).
[11] Défossez, A., Usunier, N., Bottou, L. and Bach, F. R.: Music Source Separation in the Waveform Domain, *CoRR*, Vol. arXiv:1911.13254 (2021).
[12] Hu, Y., Chen, Y., Yang, W., He, L. and Huang, H.: Hierarchic Temporal Convolutional Network With Cross-Domain Encoder for Music Source Separation, *IEEE Signal Processing Letters*, Vol. 29, pp. 1517–1521 (2022).
[13] Kong, Q., Cao, Y., Liu, H., Cho, K. and Wang, Y.: Decoupling Magnitude and Phase Estimation with Deep ResUNet for Music Source Separation, *Proc. ISMIR 2021*, pp. 342–349 (2021).
[14] Kim, M., Choi, W., Chung, J., Lee, D. and Jung, S.: KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing, *Proc. MDX 2021*, pp. 1–7 (2021).
[15] Défossez, A.: Hybrid Spectrogram and Waveform Source Separation, *Proc. MDX 2021*, pp. 1–11 (2021).
[16] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., FitzGerald, D. and Pardo, B.: An Overview of Lead and Accompaniment Separation in Music, *IEEE/ACM TASLP*, Vol. 26, No. 8, pp. 1307–1335 (2018).
[17] Gupta, C., Li, H. and Goto, M.: Deep Learning Approaches in Topics of Singing Information Processing, *IEEE/ACM TASLP*, Vol. 30, pp. 2422–2451 (2022).
[18] Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A. and Weyde, T.: Singing Voice Separation with Deep U-Net Convolutional Networks, *Proc. ISMIR 2017* (2017).
[19] Stöter, F.-R., Uhlich, S., Liutkus, A. and Mitsufuji, Y.: Open-Unmix - A Reference Implementation for Music Source Separation, *Journal of Open Source Software*, Vol. 4, No. 41, p. 1667 (online), available from ⟨https://doi.org/10.21105/joss.01667⟩ (2019).
[20] Takahashi, N., Goswami, N. and Mitsufuji, Y.: MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation, *Proc. IWAENC 2018* (2018).
[21] Jansson, A., Bittner, R. M., Ewert, S. and Weyde,

T.: Joint Singing Voice Separation and F0 Estimation with Deep U-Net Architectures, *Proc. EUSIPCO 2019* (2019).

[22] Nakano, T., Yoshii, K., Wu, Y., Nishikimi, R., Lin, K. W. E. and Goto, M.: Joint Singing Pitch Estimation and Voice Separation Based on a Neural Harmonic Structure Renderer, *IEEE WASPAA 2019* (2019).

[23] Hung, Y.-N. and Lerch, A.: Multitask learning for instrument activation aware music source separation, *Proc. ISMIR 2020* (2020).

[24] Sawata, R., Uhlich, S., Takahashi, S. and Mitsufuji, Y.: All for One and One for All: Improving Music Separation by Bridging Networks, *Proc. ISMIR 2020* (2020).

[25] Hennequin, R., Khlif, A., Voituret, F. and Moussallam, M.: Spleeter: A fast and efficient music source separation tool with pre-trained models, *Journal of Open Source Software*, Vol. 5, No. 50, p. 2154 (online), DOI: 10.21105/joss.02154 (2020). Deezer Research.

[26] Schulze-Forster, K., Doire, C. S. J., Richard, G. and Badeau, R.: Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation, *IEEE/ACM TASLP*, Vol. 29, pp. 2382–2395 (2021).

[27] Choi, W., Kim, M., Chung, J., Lee, D. and Jung, S.: Investigating U-Nets with various Intermediate Blocks for Spectrogram-based Singing Voice Separation, *Proc. ISMIR 2020* (2020).

[28] Choi, W., Kim, M., Chung, J. and Jung, S.: LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation, *Proc. ICASSP 2021*, pp. 171–175 (2021).

[29] Wang, Z., Giri, R., Isik, U., Valin, J.-M. and Krishnaswamy, A.: Semi-Supervised Singing Voice Separation With Noisy Self-Training, *Proc. IEEE ICASSP 2021* (2021).

[30] Stöller, D., Ewert, S. and Dixon, S.: Wave-U-Net: A Multi-scale Neural Network for End-to-end Audio Source Separation, *Proc. ISMIR 2017* (2017).

[31] Luo, Y. and Mesgarani, N.: Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation, *IEEE/ACM TASLP*, Vol. 27, No. 8, pp. 1256–1266 (2019).

[32] Nakamura, T. and Saruwatari, H.: Time-Domain Audio Source Separation Based on Wave-U-Net Combined with Discrete Wavelet Transform, *Proc. IEEE ICASSP 2020* (2020).

[33] Samuel, D., Ganeshan, A. and Naradowsky, J.: Meta-learning extractors for music source separation, *Proc. IEEE ICASSP 2020* (2020).

[34] Nachmani, E., Adi, Y. and Wolf, L.: Voice separation with an unknown number of multiple speakers, *Proc. ICML 2020* (2020).

[35] Prétet, L., Hennequin, R., Royo-Letelier, J. and Vaglio, A.: Singing Voice Separation: A Study on Training Data, *Proc. IEEE ICASSP 2019* (2019).

[36] Cohen-Hadria, A., Roebel, A. and Peeters, G.: Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation, *EUSIPCO 2019* (2019).

[37] Wang, Y., Stoller, D., Bittner, R. M. and Bello, J. P.: Few-Shot Musical Source Separation, *Proc. IEEE ICASSP 2022*, pp. 121–125 (2022).

[38] Kawamura, M., Nakamura, T., Kitamura, D., Saruwatari, H., Takahashi, Y. and Kondo, K.: Differentiable Digital Signal Processing Mixture Model for Synthesis Parameter Extraction from Mixture of Harmonic Sounds, *Proc. IEEE ICASSP 2022*, pp. 941–945 (2022).

[39] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I. and Bittner, R.: The MUSDB18 corpus for music separation, `https://doi.org/10.5281/zenodo.1117372`.

[40] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I. and Bittner, R.: MUSDB18-HQ - an uncompressed version of MUSDB18, `https://doi.org/10.5281/zenodo.3338373`.

[41] Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N. and Mitsufuji, Y.: Improving music source separation based on deep neural networks through data augmentation and network blending, *Proc. IEEE ICASSP 2017* (2017).

[42] Jaureguiberry, X., Vincent, E. and Richard, G.: 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).

[43] Roux, J. L., Watanabe, S. and Hershey, J. R.: Ensemble learning for speech enhancement, *Proc. WASPAA 2013* (2013).

[44] Jiang, W., Liang, S., Dong, L., Yang, H., Liu, W. and Wang, Y.: Cross-domain cooperative deep stacking network for speech separation, *Proc. IEEE ICASSP* (2015).

[45] Driedger, J. and M{" uller, M.: Extracting Singing Voice from Music Recordings by Cascading Audio Decomposition Techniques, *Proc. IEEE ICASSP 2015*, pp. 126–130 (2015).

[46] Grais, E. M., Roma, G., Simpson, A. J. R. and Plumbley, M. D.: Combining mask estimates for single channel audio source separation using deep neural networks, *Proc. Interspeech 2016* (2016).

[47] Zhang, X.-L. and Wang, D.: 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).

[48] McVicar, M., Santos-Rodriguez, R. and Bie, T. D.: Learning to Separate Vocals from Polyphonic Mixtures via Ensemble Methods and Structured Output Prediction, *Proc. IEEE ICASSP 2016*, pp. 450–454 (2016).

[49] Manilow, E., Seetharaman, P., Pishdadian, F. and Pardo, B.: Predicting Algorithm Efficacy for Adaptive Multi-cue Source Separation, *Proc. IEEE WASPA 2017*, pp. 274–278 (2017).

[50] Mitsufuji, Y., Fabbro, G., Uhlich, S., Stöter, F.-R., Défossez, A., Kim, M., Choi, W., Yu, C.-Y. and Cheuk, K.-W.: Hybrid Spectrogram and Waveform Source Separation, *Proc. Music Demixing Workshop (MDX 2021)*, pp. 1–8 (2021).

[51] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M. and Dosovitskiy, A.: MLP-Mixer: An all-MLP Architecture for Vision, *CoRR*, Vol. abs/2105.01601 (online), available from ⟨https://arxiv.org/abs/2105.01601⟩ (2021).

[52] Ba, J. L., Kiros, J. R. and Hinton, G. E.: Layer Normalization, *CoRR*, Vol. arXiv:1607.06450 (2016).

[53] Hendrycks, D. and Gimpel, K.: Gaussian Error Linear Units (GELUs), *CoRR*, Vol. arXiv:1606.08415 (2016).

[54] Tae, J., Kim, H. and Lee, Y.: MLP Singer: Towards Rapid Parallel Korean Singing Voice Synthesis, *Proc. 2021 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2021)*, pp. 1–6 (2021).

[55] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *Proc. ICLR 2015* (2015).