MUSIC SOURCE SEPARATION WITH MLP MIXING OF TIME, FREQUENCY, AND CHANNEL

Tomoyasu Nakano **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{t.nakano, m.goto}@aist.go.jp

ABSTRACT

This paper proposes a new music source separation (MSS) model based on an architecture with MLP-Mixer that leverages multilayer perceptrons (MLPs). Most of the recent MSS techniques are based on architectures with CNNs, RNNs, and attention-based transformers that take waveforms or complex spectrograms or both as inputs. For the growth of the research field, we believe it is important to study not only the current established methodologies but also diverse perspectives. Therefore, since the MLP-Mixer-based architecture has been reported to perform as well as or better than architectures with CNNs and transformers in the computer vision field despite the MLP's simple computation, we report a way to effectively apply such an architecture to MSS as a reusable insight. In this paper we propose a model called TFC-MLP, which is a variant of the MLP-Mixer architecture that preserves time-frequency positional relationships and mixes time, frequency, and channel dimensions separately, using complex spectrograms as input. The TFC-MLP was evaluated with source-to-distortion ratio (SDR) using the MUSDB18-HQ dataset. Experimental results showed that the proposed model can achieve competitive SDRs when compared with state-of-the-art MSS models.

1. INTRODUCTION

Music source separation (MSS) is the task of obtaining individual source signals — such as vocals, drums, and bass — from real music acoustic signals. This is an essential technique for various applications, including music information retrieval and music listening interfaces, where the characteristics of individual sound sources are analyzed and utilized. MSS has actually been 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 achieve audio-to-lyrics alignment [9,10], to create music mashups [11], to separate sources for music education [12], and to estimate compatibility between vocals and accompaniment [13].

Currently, the mainstream approaches for MSS use deep neural networks [14, 15], and their performance is improving year by year. For their performance comparison to measure the improvement, MUSDB18 [16] had been used as the common standard data set for the four target sound sources (Vocals, Drums, Bass, and Other). Then MUSDB18-HQ [17], an extended frequency bandwidth version of MUSDB18, was released and has been used for recent evaluations.

As for the current state-of-the-art MSS models, the top-ranked models [18, 19] in the Music Demixing (MDX) Challenge 2021 [20] and the two models presented in two arXiv papers in 2022 [21, 22] have an average SDR (source-to-distortion ratio) over 7 dB for the four sources when using MUSDB18-HQ as training, validation, and test data. These models are explained in the next section.

Such deep MSS models can be classified in terms of the type of input and output used for separation and the type of architecture. The input and output of the models are selected from waveforms, amplitude spectrograms, complex spectrograms, phase spectrograms, etc. The architecture of the models is mainly selected from ResNet, DenseNet, U-Net, and Transformer and is used with layers of Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). Simpler architectures based on multilayer perceptrons (MLPs), however, have not been used in state-of-the-art MSS models.

Meanwhile, in the field of computer vision, high-performance architectures based on MLPs have recently been proposed and reported to perform as well as or better than architectures using CNNs or transformers [23,24]. In addition, those simpler MLP-centric architectures have also been applied in audio-related research, with cases of singing voice synthesis [25] and speech enhancement [26]. However, such MLP-centric architectures have not been applied in the MSS domain, though fully connected layers and MLPs have been used for linear transformation such as embedding and expansion. Since we believe that new perspectives, in addition to the development of established methodologies, are important for the advancement of the research field, this paper investigates how MLP-based architectures can be effectively leveraged for MSS.

As such a modern MLP-centric high-performance architecture, Tolstikhin *et al.* proposed the *MLP-Mixer* [23] that applies MLPs to a $T_p \times C_p$ matrix to estimate the

© T. Nakano and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** T. Nakano and M. Goto, "Music source separation with MLP mixing of time, frequency, and channel", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Table 1. SDRs in MUSDB18-HQ for the state-of-the-art models and our proposed TFC-MLP. The "Avg." column means the average of the SDR results for the four sources. The "Per-source" column means that the per-source adjustment/tuning has been implemented. The "Extra" column indicates the number of songs added as extra training data, and † means that only mixed sounds were added. Models with "*" are evaluated in MUSDB18 [16], and SDRs for models with "**" are recalculated in MUSDB18-HQ using the pretrained model. A **bold** font indicates the maximum value at each source.

Model			Test SDR in dB				
Name	Per-source	Extra	Avg.	Vocals	Drums	Bass	Other
KUIELab-MDX-Net (w/o Demucs) [18]*	✓		7.28	8.91	7.07	7.33	5.81
TFC-MLP (ours)			7.30	8.91	7.18	6.96	6.14
KUIELab-MDX-Net [18]**	✓		7.48	8.97	7.20	7.83	5.90
Hybrid Transformer Demucs [22]			7.52	7.93	7.94	8.48	5.72
Hybrid Demucs [19]			7.64	8.35	8.12	8.43	5.65
Band-Split RNN [21]	✓		8.24	10.01	9.01	7.22	6.70
TFC-MLP (ours)		120	7.78	9.68	7.75	7.23	6.46
Hybrid Demucs [19]		800	8.34	8.75	9.31	9.13	6.18
Hybrid Transformer Demucs [22]		150	8.49	8.56	9.51	9.76	6.13
Hybrid Transformer Demucs [22]		800	8.80	8.93	10.05	9.78	6.42
Band-Split RNN [21]	✓	1750†	8.97	**10.47**	10.15	8.16	**7.08**
Hybrid Transformer Demucs [22]	✓	800	9.00	9.20	10.08	10.39	6.32
Sparse HT Demucs [22]	✓	800	**9.27**	9.37	**10.83**	**10.47**	6.41

class of an image. The matrix is obtained by dividing the image into patches and embedding each patch into a T_p-dimensional vector, which is called a token, and the number of tokens is the channel dimension C_p. Here, a *token-mixing MLP*, a full connection within an individual token, and a *channel-mixing MLP*, a full connection between tokens (*i.e.,* channel direction), are applied alternately. Given sufficient training data, MLP-Mixer was shown to perform as well as or better than CNNs or transformers.

By extending this MLP-Mixer, for the image reconstruction task, Mansour *et al.* proposed the *Image-to-Image Mixer* [24] that performs better when trained with fewer images than the original MLP-Mixer. The Image-to-Image Mixer transforms images as a 3D tensor ($W \times H \times C$) instead of the 2D matrix ($T_p \times C_p$) and preserves the relative positions of patches to induce a bias towards natural images. In other words, the token-mixing MLP and the channel-mixing MLP are split into three processes of *width-mixing MLP*, *height-mixing MLP* and *channel-mixing MLP*. This also keeps the total number of trainable parameters low, since the size of each dimension of the 2D matrix (*i.e.,* T_p and C_p) obtained by transformation from the 3D tensor is relatively large.

In investigating such MLP-based architectures in the MSS domain, we decided to use a complex spectrogram, which is a reasonable representation for MSS, as the input. Since the size of its complex time-frequency representation is larger than the size of typical images in the computer vision domain, we apply the memory-efficient Image-to-Image Mixer to MSS and report its experimental results.

2. RELATED WORK

As described in Section 1, the state-of-the-art models [18, 19, 21, 22] in the MSS study show that the average SDR score for the four source separations exceeds 7 dB in the evaluation using MUSDB18-HQ. The SDRs for each of the four sources based on these models are shown in Table 1.

KUIELab-MDX-Net was proposed by Kim *et al.* [18]. It is an architecture that combines an extended version of TFC-TDF-U-Net [27], which separates in the time-frequency domain (*i.e.,* complex spectrogram), and Demucs [28], which separates in the time domain (*i.e.,* waveform). Each source signal i separated by those subnetworks is mixed using source-dependent weights w_i. 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 [20] [1]. In KUIELab-MDX-Net, to improve performance, a mechanism called *Mixer* was used to remix the music acoustic signal with the separated signal by using the 1x1 convolution, and different FFT frame sizes were used for each sound source by applying a frequency cut-off trick [20].

TFC-TDF-U-Net used in the KUIELab-MDX-Net is a variant of U-Net architecture that combines the time-frequency convolutions (TFC) block with the time-distributed fully-connected networks (TDF) block. Here, TFC is a dense block of 2D CNNs, and TDF is a block that extracts nonlocal features along the frequency axis, such as correlations between harmonics, by fully connecting the entire frequency range of a single frame of the spectrogram. TDF was inspired by the Frequency Transformation Block (FTB) proposed by Yin *et al.* [29] and was introduced specifically to help separate singing voices [27]. As the other component of KUIELab-MDX-Net, Demucs [28] is a U-Net encoder/decoder structure with waveforms as input and BiLSTM applied to the innermost layer between the encoder and decoder to provide long-range context.

Hybrid Demucs was proposed by Défossez *et al.* [19]. It is a bi-U-Net encoder-decoder model that combines 1D convolution in the time domain and along the frequency

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

axis in the complex time-frequency domain. BiLSTM and local attention were used in the innermost layer, and residual branches, group normalization [30] and GELU were introduced. In addition, better generalization and stability were achieved by penalizing the largest singular value in each layer [31], and overall performance was improved by bagging multiple models.

Band-Split RNN was proposed by Luo *et al.* [21]. It is a state-of-the-art spectrogram-based model that uses complex spectrograms as input and output. By splitting the complex spectrogram input into subbands specifically designed for each sound source, the intrinsic properties and patterns of each source signal are utilized. For a 3D tensor representing the time dimension, frequency dimension, and subband dimension, similar to the Dual-path RNN [32], a sequence-level RNN is first applied across the time dimension, then a band-level RNN is applied across the band dimension. In fact, for Vocals, results showed that the modified utterance-level SDR can be improved by more than 2 dB by setting the bandwidth appropriately. In addition, semi-supervised fine tuning was performed by using pseudo-targets separated from the mixtures using a pretraining model in order to make effective use of songs with only mixtures (1750 songs). The Band-Split RNN currently achieves a state-of-the-art SDR of 8.24 dB in the evaluation using only MUSDB18-HQ as training data.

Hybrid Transformer Demucs was proposed by Rouard *et al.* [22]. The innermost layer, called the bottleneck, of the Hybrid Demucs [19] is replaced by a cross-domain transformer encoder (*i.e.,* time domain and time-frequency domain) that uses self-attention within each domain and cross-attention across domains. Compared to the Hybrid Demucs, the performance is lower when trained with MUSDB18-HQ only, but the SDR was 0.46 dB better when 800 additional training songs were used. Sparse HT Demucs [22], which extended the receptive field using a sparse attention kernel, achieved a state-of-the-art result of 9.27 dB SDR by fine-tuning for each source.

As described above, the MLP-centric architecture has never been applied in state-of-the-art MSS models where the average SDR exceeds 7 dB on MUSDB18-HQ.

3. TFC-MLP

This paper proposes *Time-Frequency-Channel-MLP (TFC-MLP)*, which is a model that leverages the Image-to-Image Mixer architecture [24] to separate music sources using a complex spectrogram as input. This is realized by replacing the height, width, and color (RGB) in the image with the time, frequency, and channel in the complex spectrogram, respectively. In other words, TFC-MLP has a structure that alternates mixing in the time, frequency, and channel dimensions. In this way, we expect to be able to take into account the nonlocal structure. Especially with respect to the frequency dimension, we expect to extract nonlocal relationships along the frequency axis, such as harmonic structures, by connecting the entire frequency range of the spectrogram, as in the FTB [29] and the TDF [27].

The process of the TFC-MLP model is shown in Figure 1. The complex spectrogram $\mathbf{X}_{\text{STFT}} \in \mathbb{C}^{F_0 \times T_0 \times 2}$ of a 2-channel (stereo) mixture signal is first converted to a 4-channel 3D tensor $\mathbf{X}_{\text{CaC}} \in \mathbb{R}^{F_0 \times T_0 \times 4}$ with the real and imaginary parts represented as channels, a.k.a. complex-as-channels (CaC) [27]. Here T_0 is a fixed length. Then, as with the MLP-Mixer and Image-to-Image Mixer as well as the Vision Transformer (ViT) [33], the patch embedding is first performed using C linear weights of size $P_T \times P_F \times 4$ (Figure 2). This reduces the frequency dimension F_0 to F_0/P_F and the time dimension T_0 to T_0/P_T, which reduces the matrix size and memory consumption in the following frequency-mixing MLP and time-mixing MLP. To compensate for the information loss due to decreasing the resolution in the time-frequency plane and the loss of continuity in the time-frequency direction, we increase the dimension in the channel direction.

The embedded tensor then passes through N MLP-Mixer layers. As shown in Figure 3, each MLP-Mixer layer mixes the tensor with the frequency dimension, then the time dimension, and finally the channel dimension. Such mixing is passed through an MLP consisting of a linear layer, a GELU nonlinearity, and another linear layer, and output without changing the tensor size. The dimension of the tensor at the input/output layer of the MLP is kept constant, and the dimension at the hidden layer is adjusted by multiplying it by a factor f depending on the input dimension. Skip connections and channel layer normalization have also been added to help with the optimization. Here, following the implementation of the Image-to-Image Mixer [24], skip connections are placed before and after the two mixing steps of frequency and time ("Type A"). In addition to that, a version with skip connection and layer normalization added before time-mixing MLP was also implemented ("Type B"). After mixing the time, frequency, and channel dimensions, patch expansion is used to restore the number of time and frequency dimensions for inverse transformation into waveforms and also to match the channel dimension to the number of separated sources.

4. EVALUATION

The proposed TFC-MLP was evaluated using the MUSDB18-HQ dataset [17]. 86 songs were used as training data, 14 songs as validation data, and 50 songs as test data. The music acoustic signals were stereo with a sampling frequency of 44.1 kHz, and four sound sources – "Vocals," "Drums," "Bass," and "Other" – were used for separation. Separation performance was evaluated by calculating SDR using the *museval* Python package[2]. As in most previous studies ([19, 21], etc.), the SDR of each source was 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.

4.1 Experimental setting

The proposed model was optimized using Adam [34] for the L1 loss between its separated signals and the ground-

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

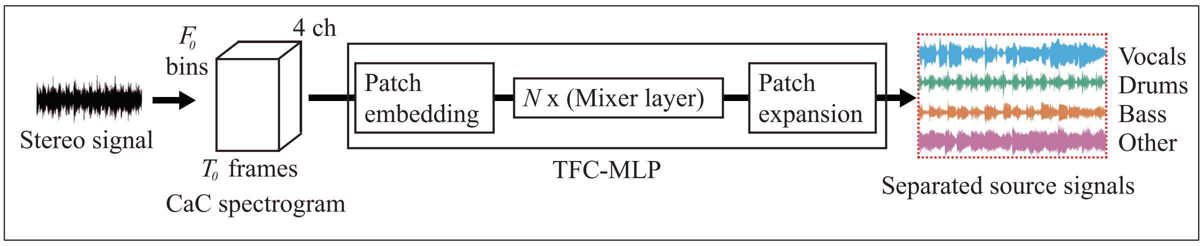

Figure 1. Overview of the TFC-MLP model.

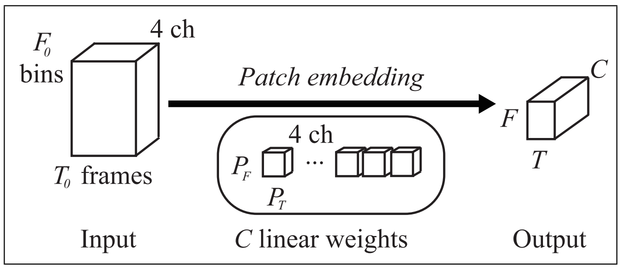

Figure 2. Patch embedding. After dividing the CaC tensor of size $F_0 \times T_0 \times 4$ into patches of size $P_F \times P_T \times 4$, each patch was linearly transformed into $1 \times 1 \times C$ to obtain a 3D tensor of size $F_0/P_F \times T_0/P_T \times C$ (*i.e.,* $F \times T \times C$). This can be implemented as a nonoverlapping CNN.

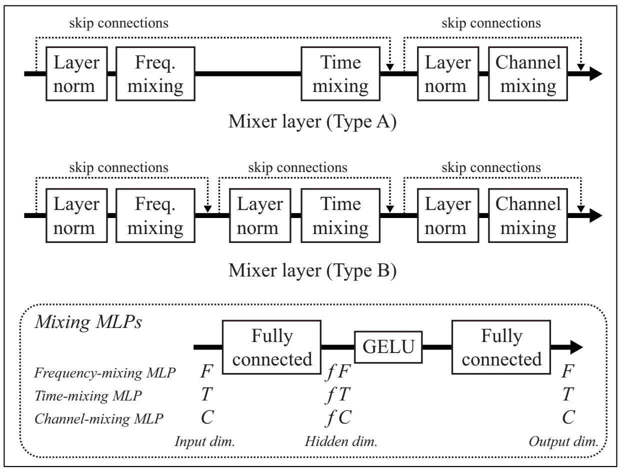

Figure 3. The Mixer layer contains one frequency-mixing MLP, one time-mixing MLP, and one channel-mixing MLP. Before each MLP, a transposition is performed to apply the MLP to the frequency dimension F, the time dimension T, and the channel dimension C. Transposition is also performed before every layer normalization to normalize along the channel dimension C.

truth source signals. The Adam parameters were set as no weight decay, learning rate 0.0003, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The waveform for calculating the CaC spectrogram to be input to the model was normalized so that the mean amplitude of the music acoustic signal was 0 and the standard deviation was 1. The training was distributed across multiple GPUs, with a batch size of 4 on each GPU. In training, we used data augmentation techniques [19], including pitch shift and tempo stretch, randomly swapping channels, random sign flip, random scaling of amplitude, and remixing of stems within one batch.

4.2 Base hyperparameters

Due to the many hyperparameters required for the building of the proposed TFC-MLP model, hyperparameters related to the short-time Fourier transform (STFT) were first determined through preliminary experiments. To obtain $\mathbf{X}_{\mathrm{CaC}}$, the STFT frame size (*i.e.,* FFT size) was chosen as 4096, which had the highest SDR when compared among 512, 1024, 2048, and 4096. As Kim *et al.* [18] also stated, the performance tended to increase with larger FFT size. Therefore the frequency dimension F_0 is 2048. Related to the FFT size, the STFT hop size was investigated among 128, 256, 512, and 1024, and 1024 was selected. Also related to the FFT size and STFT hop size, the number of time frames T_0 in the complex spectrogram was selected as 128 from 32, 64, 128, 256, and 512.

The number of time frames T_0 of 128 based on an FFT size of 4096 and an STFT hop size of 1024 is input to the model, thus the waveform size required for this is equal to $T_w = 1024 \times 128 - 1$ (about 3 seconds). Therefore, it is necessary to train the model while cutting out an approximately 3-second segment of the waveform, and the shift size of 16384 (about 0.37 seconds) was used. Luo *et al.* [21] found that the shorter the shift, the better the modified utterance-level SDR, and they used 0.5 seconds. The above 16384 is short enough compared to 0.5 seconds and can be considered sufficient in this study.

As hyperparameters regarding the model, the dimensions C of the patch embedding were investigated by us. Specifically, 128 and 256 were investigated and 256 was selected. As hyperparameters related to the Mixer layer, 8 and 16 were investigated as the number of layers N and 16 was selected, and the parameter f for adjusting the number of dimensions of the hidden layer of each MLP was selected as 4 from 1, 2, and 4. Dropout in the MLP [25] and skip connections before and after the Mixer layer were also investigated, but they have not yielded better results.

4.3 Experiment

Using the set of hyperparameters determined in Section 4.2 as a basis, we report the SDR scores obtained under the following various conditions for the other hyperparameters.

- Investigation of the results of complex spectrogram loss [21].

Table 2. SDRs for different hyperparameters of the proposed model. The "Seed" column indicates random seeds. In the "Loss" column, "W" indicates that only waveform loss was used, and "W+S" indicates that complex spectral loss was added. The "Epoch" column indicates the number of epochs with the smallest validation loss and the specified number of epochs, where "*" means that validation loss was not considered. Note that "†" means the cases where the number of epochs with the smallest validation loss did not change when training beyond 200 epochs (*i.e.,* the same model was used for the evaluation). A **bold** font indicates the maximum value at each source, both without and with extra training songs.

TFC-MLP						Test SDR in dB				
Type	Seed	(P_F, P_T)	Loss	Epoch	Extra	Avg.	Vocals	Drums	Bass	Other
A	seed 1	$(4, 4)$	W	134† / 200		**7.30**	8.91	**7.18**	6.96	6.14
A	seed 2	$(4, 4)$	W	166† / 200		7.26	8.84	7.07	6.89	**6.22**
A	seed 3	$(4, 4)$	W	109 / 200		6.97	8.32	6.98	6.57	5.99
A	seed 3	$(4, 4)$	W	283 / 300		6.93	8.49	6.80	6.55	5.87
B	seed 1	$(4, 4)$	W	190 / 200		7.17	**8.92**	6.95	6.83	5.96
B	seed 2	$(4, 4)$	W	191 / 200		7.02	8.59	6.78	6.68	6.01
B	seed 3	$(4, 4)$	W	157 / 200		6.95	8.56	6.58	6.73	5.91
A	seed 1	$(2, 2)$	W	189 / 200		7.13	8.58	7.02	**6.99**	5.91
A	seed 1	$(1, 1)$	W	175 / 200		6.38	7.40	6.33	6.49	5.30
A	seed 1	$(4, 4)$	W+S	197 / 200		6.83	8.76	6.60	6.14	5.83
A	seed 1	$(4, 4)$	W+S	253 / 300		6.72	8.39	6.69	6.02	5.79
A	seed 1	$(4, 4)$	W	142 / 200	120	7.71	9.42	7.66	**7.37**	6.39
A	seed 1	$(4, 4)$	W	200* / 200	120	**7.78**	**9.68**	**7.75**	7.23	**6.46**

Table 3. The number of model parameters and the average Real Time Factor (RTF) value. The Hybrid Transformer Demucs is denoted as "HT Demucs". Column "GPU" shows the RTF with a single GPU, and column "CPU" shows the RTF under the condition without a GPU.

Model	GPU	CPU	Params.
Hybrid Demucs	0.14	1.87	83.9 M
HT Demucs	0.17	2.55	26.9 M
TFC-MLP	0.51	12.28	43.2 M

- Investigation of patch size (P_F, P_T) between $(4, 4)$, $(2, 2)$, and $(1, 1)$, using $(4, 4)$ as the basis, halving the FFT size and time dimension T for $(2, 2)$, and halving it further for $(1, 1)$. For $(1, 1)$, the STFT hop size was set to 512 since the FFT size is 1024.
- The number of epochs for training was set to 200 or 300, and models with the smallest validation loss within each epoch-condition were evaluated.
- The results of adding 120 full-length songs (sung in English) to the training data were evaluated.
- For the basic parameter condition, we trained three times with different random seeds.

In the test phase, the model with the smallest validation loss was used for evaluation in each training condition. The signal separated by the proposed model was divided into segments of fixed length T_w with shift width $T_w/4$, which were weighted overlap-added to obtain the final signal. In addition, the *shift trick* [28] was performed 10 times.

4.4 Results and discussions

In addition to the evaluation of SDRs using the MUSDB18-HQ test set, the number of parameters and the Real Time Factor (RTF) will also be discussed, as file size and the time required for separation may be important in some situations depending on how the MSS model is used.

4.4.1 SDRs

The experimental results are shown in Table 2. The highest SDR score, up to 7.30 dB, was obtained for the Type A model using the patch size of $(4, 4)$ and waveform L1 loss in addition to the base hyperparameters. This was not significantly higher than the state-of-the-art values shown in Table 1, but close performance was achieved.

Focusing on the results for each source with respect to our TFC-MLP, as shown in Table 1, the SDRs for Vocals and Other were 8.91 dB and 6.14 dB, respectively, which were higher than the 8.35 dB and 5.65 dB SDRs for the Hybrid Demucs and the 7.93 dB and 5.72 dB SDRs for the Hybrid Transformer Demucs. This model's SDR score for Other also exceeded the one obtained using KUIELab-MDX-Net, 5.90. Here, in comparison to the KUIELab-MDX-Net results without waveform information (*i.e.,* excluding processing by Demucs), it is possible that similar or better performance was obtained for Drums and Vocals, although an exact comparison cannot be made because the test data is different (*i.e.,* MUSDB18 was used). In comparison to the Band-Split RNN results, TFC-MLP could not yield a higher SDR for all sound sources. However, the current TFC-MLP does not include a source-specific framework, so addressing this issue is a future challenge.

As for the patch size, the FFT size and other conditions were different due to memory capacity. Therefore, although exact comparisons are not possible, the best results were obtained for $(4, 4)$ in the current results. However, additional study is needed for $(2, 2)$, since it gave results similar to those given by $(4, 4)$. As for complex spectrogram loss, its SDR was slightly lower than that of all conditions using only waveform loss. Not only comparisons using the real and imaginary parts of the complex numbers, but also losses based on amplitude and phase could be considered.

We also showed that the performance of the models was further improved by using additional training data. As shown in Table 1, compared to the current world's best model Sparse HT Demucs with 800 songs added as the training data, we obtained competitive results with an SDR of 9.68 dB for the Vocals and 6.46 dB for the Others.

4.4.2 RTF and the number of parameters

As comparison, models were trained for each of the Hybrid Demucs and Hybrid Transformer Demucs. Based on the published source codes, a model parameter setting "80a68df8" [3] was used for Hybrid Demucs, and the default parameter setting was used for Hybrid Transformer Demucs. The same implementation for audio synthesis was used for all TFC-MLP, Hybrid Demucs, and Hybrid Transformer Demucs. We used the 50 songs in the MUSDB18-HQ test set to obtain the average of their RTFs.

Table 3 shows the results. The RTF values of TFC-MLP were slower than the other two models. TFC-MLP had an RTF lower than 1.0 (faster than real time) when a GPU was used, but the computation time was long without GPU. This could be due to the large size of the time-frequency spectrogram. As for the number of model parameters, TFC-MLP had more parameters than Hybrid Transformer Demucs, but fewer parameters than Hybrid Demucs.

4.4.3 Comparison with the state-of-the-art models

The proposed TFC-MLP model has some similarities to the state-of-the-art MSS models, which potentially have led to the competitive performance achieved.

- The frequency-mixing MLP is similar to the full connection of frequency dimensions in TDF [27] and the band-level RNN applied across band dimensions in Band-Split RNN [21].
- The time-mixing MLP is similar to the sequence-level RNNs applied across time dimensions in Band-Split RNN [21].
- Increasing the number of the channel dimensions through patch embedding and increasing the number of hidden layer dimensions in MLP are techniques that are usually used regardless of MSS. For MSS, the improvement is potentially related to the increase in channel dimensionality in the encoder part, such as Hybrid Transformer Demucs [22].
- The extra training data improved performance in the state-of-the-art models, and we confirmed the performance improvement with extra training data in TFC-MLP as well.

On the other hand, the following are included in the existing state-of-the-art MSS models but not currently included in our TFC-MLP. They have the potential to improve performance when applied in the future.

- As presented by Défossez *et al.* [19] and Kim *et al.* [18], a hybrid approach that also considers waveforms could improve performance.
- As Kim *et al.* [18], Luo *et al.* [21], and Rouard *et al.* [22] have shown, the introduction of source-specific techniques, such as band splitting, could improve separation performance.
- Deep learning techniques such as model selection methods (*e.g.,* exponential moving average), training stabilization (*e.g.,* singular value decomposition and sparsification), and the introduction of a learning rate scheduler could further improve performance.

Finally, to the best of our knowledge, there are no studies that mix the channel dimension with the time and frequency dimensions as in TFC-MLP. Such a mixer layer used in the TFC-MLP architecture has the advantage of reducing the overall memory usage compared to the original MLP-Mixer, just as the Image-to-Image Mixer reduced the memory usage. This reusable insight of mixing the channel dimension separately could be useful for other studies that have dealt with the time and frequency dimensions so far, but could be extended to the channel dimension.

4.4.4 Future directions

As future work, we plan to improve the performance of TFC-MLP by incorporating the ideas discussed above and further exploring more optimal hyperparameters. For example, increasing the FFT window size by utilizing frequency cut-off trick [20] is expected to improve the performance. Automatic optimization of hyperparameters could also be incorporated.

Future work will also include the visualization of the inside of TFC-MLP for analysis. We could visualize the linear weights used in the patch embedding by converting them back to complex numbers and then calculating their amplitudes. The visualized results could allow us to analyze what local patterns the model is focusing on. However, when we tried it, it was difficult to understand the behavior of the mixer layer due to the mixing of real and imaginary parts during the patch embedding. We are therefore interested in using the amplitude and phase (group delay) instead of the real and imaginary parts so that we can analyze the model in a more comprehensive way.

5. CONCLUSION

This paper has described a new MSS architecture called TFC-MLP that uses complex spectrograms as input. Our contributions are summarized as follows:

(1) We proposed a simpler MLP-centric MSS architecture that achieves competitive performance compared to state-of-the-art models.
(2) We reported on some hyperparameter searches that will be useful for other researchers exploring this type of architecture.
(3) We discussed the similarities and differences between the state-of-the-art models and TFC-MLP, and suggested directions for future research.

[3] `https://github.com/facebookresearch/demucs/blob/main/docs/training.md`

6. ACKNOWLEDGMENTS

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

7. REFERENCES

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

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

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

[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," in *Proc. the 9th International Conference of Music Information Retrieval (ISMIR 2008)*, 2008, pp. 133–138.

[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," in *Proc. the 2020 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020)*, 2020, pp. 1979–1989.

[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," in *Proc. the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2020–2024.

[9] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[10] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *Proc. the 23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, 2022, pp. 187–195.

[11] J. Huang, J.-C. Wang, J. B. L. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proc. the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021, pp. 187–195.

[12] E. Cano, G. Schuller, and C. Dittmar, "Pitch-informed solo and accompaniment separation towards its use in music education applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 23, pp. 1–19, 2014.

[13] 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. 101 994–102 003, 2021.

[14] 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.

[15] 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.

[16] 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.

[17] ——, "MUSDB18-HQ - an uncompressed version of MUSDB18," https://doi.org/10.5281/zenodo.3338373.

[18] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "KUIELab-MDX-Net: A two-stream neural network for music demixing," in *Proc. Music Demixing Workshop 2021 (MDX 2021)*, 2021, pp. 1–7.

[19] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proc. Music Demixing Workshop 2021 (MDX 2021)*, 2021, pp. 1–11.

[20] 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," *Front. Sig. Proc.*, 2022.

[21] Y. Luo and J. Yu, "Music source separation with band-split rnn," *CoRR, arXiv:2209.15174*, pp. 1–10, 2022.

[22] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," *CoRR, arXiv:2211.08553*, pp. 1–5, 2022.

[23] 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," in *Proc. the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, pp. 24 261–24 272.

[24] Y. Mansour, K. Lin, and R. Heckel, "Image-to-Image MLP-Mixer for image reconstruction," *CoRR, arXiv:2202.02018*, pp. 1–15, 2022.

[25] J. Tae, H. Kim, and Y. Lee, "MLP Singer: Towards rapid parallel korean singing voice synthesis," in *Proc. the 2021 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2021)*, 2021, pp. 1–6.

[26] H. Song, M. Kim, and J. W. Shin, "Speech enhancement using mlp-based architecture with convolutional token mixing module and squeeze-and-excitation network," *IEEE Access*, vol. 10, pp. 119 283–119 289, 2022.

[27] W. Choi, M. Kim, J. Chung, D. Lee, and S. Jung, "Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation," in *Proc. the 21st International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 192–198.

[28] 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.

[29] D. Yin, C. Luo, Z. Xiong, and W. Zeng, "PHASEN: A phase-and-harmonics-aware speech enhancement network," in *Proc. the The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, 2020, pp. 9458–9465.

[30] Y. Wu and K. He, "Group normalization," in *Proc. the 15th European Conference on Computer Vision (ECCV 2018)*, 2018, pp. 3–19.

[31] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *CoRR, arXiv:1705.10941*, pp. 1–12, 2017.

[32] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 46—-50.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. the Ninth International Conference on Learning Representations (ICLR 2021)*, 2021.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015, pp. 1–15.