EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH Open Access

Stripe-Transformer: deep stripe feature learning for music source separation

Jiale Qian[1], Xinlu Liu[1], Yi Yu[2] and Wei Li[1,3*]

Abstract

Music source separation (MSS) is to isolate musical instrument signals from the given music mixture. Stripes widely exist in music spectrograms, which potentially indicate high-level music information. For example, a vertical stripe indicates a drum time and a horizontal stripe indicates a harmonic component such as a singing voice. These stripe features actually affect the performance of MSS systems, which has not been explicitly explored by previous MSS studies. In this paper, we propose stripe-Transformer, a deep stripe feature learning method for MSS with a Transformer-based architecture. Stripe-wise self-attention mechanism is designed to capture global dependencies along the time and frequency axis in music spectrograms. Experimental results on the Musdb18 dataset show that our proposed model reaches an average source-to-distortion (SDR) of 6.71dB on four target sources, achieving state-of-the-art performance with fewer parameters. And the visualization results show the capability of the proposed model to extract beat and harmonic structure in music signals.

Keywords Music source separation, Self-attention, Transformer, Stripe embedding

1 Introduction

Music source separation (MSS) is an essential technology in music information retrieval (MIR), with the aim of recovering one or more target musical sources from the mixture. The target musical sources usually refer to some musical instruments such as bass, drums, and singing voice. The mixture refers to combinations of source signals.

MSS has an extensive range of applications, such as music remixing [1] and accompaniment extraction for karaoke systems [2]. It can also be used as a preprocessing technique for other MIR tasks [3–5]. When the background accompaniment is removed, the results of some algorithms such as singer identification [6], vocal melody extraction [7], music emotion recognition [8], and query-by-humming [9] show promising improvements. Some other MIR studies also use source separation as a joint optimization target [10–14] to achieve more effective performance.

This task has been challenging for years due to the complexity of multi-source modeling and other interference factors such as background noise and reverberation. Generally, the solution to this problem can be divided into two categories according to the processing domain of the method: waveform domain [15–18] and spectrogram domain [19–31]. Spectrogram-based methods model on the spectrograms generated from the short-time Fourier transform (STFT) rather than the raw input waveform, which will be further discussed in this paper. Spectrogram-based MSS mainly includes spectral-decomposition-based [19, 20], pitch-based [21–23], repeating-based [24–26], and deep-neural-network (DNN)-based methods [27–31].

Recently, with the rapid development of deep learning, the DNN-based methods achieved more competitive

*Correspondence:
Wei Li
weili-fudan@fudan.edu.cn
[1] School of Computer Science and Technology, Fudan University, Shanghai, China
[2] Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo, Japan
[3] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China

results in MSS, using various of neural networks such as CNNs [32–34], RNNs [35, 36], and Transformers [37, 38]. These networks were trained on multi-track music datasets to learn the pattern of separating various kinds of instruments and singing voices.

However, most previous DNN-based MSS studies do not explicitly explore the stripe features widely existed in music spectrograms. Considering the existence of harmonics for melody-based instruments, there are many horizontal stripes parallelly located in integral multiples of fundamental frequency f_0 [39], which can be found in the "vocals" and "bass" spectrograms shown in Fig. 1. For example, given f_0 is 200Hz, there are corresponding harmonic components in places around 400Hz, 600Hz, etc. On the other hand, vertical stripes also appear in spectrograms when rhythmic instruments such as drum kits are played [26], as presented in the "drums" spectrogram.

The aim of our study is to appropriately process these unique characteristics of musical spectrograms for music source separation. Considering the excellent performance of deep neural networks in recent MIR developments, we choose the DNN-based architecture to process stripe features of music spectrograms.

Our contributions in this paper mainly include the following four aspects: (1) In the task of MSS, we first propose to combine U-Net architecture with the Transformer backbone network. (2) In the proposed model, high-level spectral feature maps are modeled as sequences of horizontal or vertical stripes. We design a stripe-wise self-attention (SiSA) module, a novel attention mechanism to capture long-term dependencies within and between these stripes. (3) Under the optimal experimental setting, the proposed method can achieve the state-of-the-art (SOTA) results on the Musdb18 dataset with fewer parameters. (4) We present a visual analysis for attention maps of stripe features and reconstructed spectrograms, which shows that the proposed model can better extract stripe features such as beat and harmonic structure in the music spectrograms.

2 Related works

Recently, the SOTA algorithms for music source separation are mostly based on deep neural networks. This section introduces the existing MSS methods which use the relevant neural networks involved in our proposed method.

2.1 CNN-based methods

Convolutional neural network (CNN) was initially proposed for image classification [32]. The convolution layer slides different convolution kernels on the input image with certain linear operations. This operation with the strategy of sharing parameters significantly reduces the model parameters and can better extract local features.

In the audio and music source separation task, CNNs also show effective performance. Chandna et al. proposed a network based on CNN for audio source separation [40]. Takahashi et al. proposed a DenseNet-based network which introduced connections through multiple feature maps through down-sampling and up-sampling

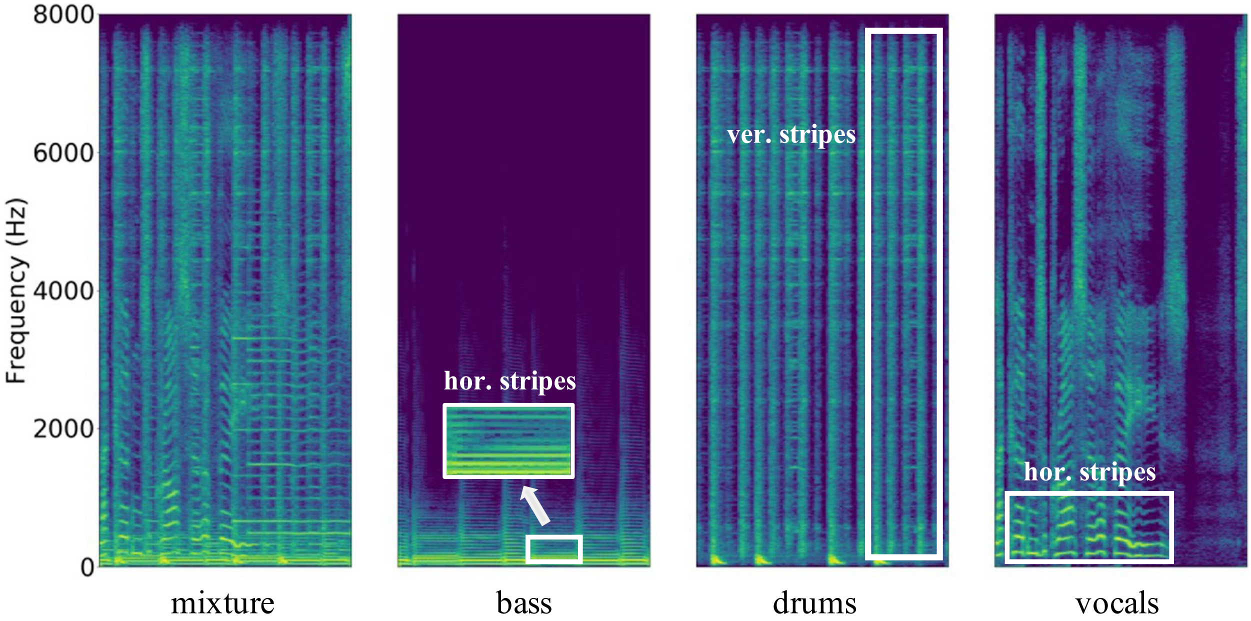


Fig. 1 Spectrograms of an example music piece and its source signals, in which horizontal and vertical stripes are highlighted

layers [41], with considerable performance improvement using fewer parameters. Stacked Hourglass Network was proposed to capture features at both coarse and fine resolution, with CNNs as the backbone network [42]. Kong et al. proposed the deep ResUNet using more stacked residual CNN layers with skip connections [29], which achieved SOTA performance using the spectrogram-based method.

2.2 U-Net-based methods

U-Net was initially proposed for medical image segmentation [43], which adopted a U-shaped encoder-decoder structure. Skip connections are introduced to connect the convolutional layers of the same resolution between the encoder and decoder. In this way, shallow features and deep semantic features are fused, which significantly boosts the performance of image segmentation.

Jansson et al. first applied U-Net to singing voice separation [28], which surpassed the SOTA method at that time in both subjective and objective indicators. Since then, there has been a series of MSS works based on this model architecture [12, 29, 44, 45]. Choi et al. verified various types of intermediate blocks that can be used in the U-Net architecture [46]. Wave-U-Net was proposed as a time domain method for audio source separation method adapted from U-Net architecture [15, 47].

2.3 Transformer-based methods

Recently, Transformers have been widely applied in the area of natural language processing [37, 38], image processing [48, 49], and audio processing [50, 51]. Transformers with the self-attention mechanism can capture long-term dependencies and highlight essential features in a parallel computation pattern. For the music source separation task, Li et al. proposed a sliced attention-based neural network, which showed the effective performance of the self-attention mechanism [30]. Yu et al. proposed a pure spectral-temporal Transformer-based encoder that outperformed previous singing voice separation methods [52].

3 Method

This section introduces the overall architecture of the proposed model and the details of its main components. The stripe-Transformer block and stripe-wise self-attention mechanism are further explained.

3.1 Overall architecture

As presented in Fig. 2, we design our proposed model according to the SIMO (single-input-multi-output) architecture [53], in which the single-input refers to the input mixture and multi-outputs refer to spectrograms of target sources. In terms of the model architecture, we use the U-Net-like structure, with the consideration of the impressive performance of this symmetric structure for source separation tasks.

It is difficult to directly learn global information from low-level feature maps, and the calculation complexity will be unbearable if the Transformer module directly models on the input spectrogram with a relatively large frequency dimension (1536 frequency bins in our experimental setting). We first down-sample the spectral feature maps by using the convolution layer with the stride of 2 on the frequency dimension. Each down-sampling convolutional layer is followed by a residual CNN block. The encoded feature maps can be reduced to 192 frequency bins using three down-sampling CNN blocks. Then, the stripe feature learning module is placed at the bottleneck part of the U-Net structure to process multi-scale feature representations. The outputs of the stripe feature learning module are then passed into the convolutional decoder. Skip connections exist between spectral feature maps of down-sampling and up-sampling processes.

3.2 Residual CNN block

Residual CNN blocks are placed at the encoder and decoder, which can focus on local regions to recover high-resolution details. The structure of a residual CNN block is presented in Fig. 2b. It consists of two convolutional layers, in which each layer follows a LeakyReLU activation and a batch normalization layer. And one more convolution layer of 1×1 kernel connects the input and the output of the main branch.

3.3 Stripe-Transformer block

Stripe-Transformer block is used to capture dependencies of horizontal and vertical stripes in multi-scale feature representations. The structure of a stripe-Transformer block is presented in Fig. 2c, which mainly consists of a stripe-wise self-attention (SiSA) module, a squeeze-and-excitation (SE) module, and a mixed-scale convolutional FFN (MixCFN).

The SiSA module is an attention-based dual-path network, which consists of two branches for processing horizontal and vertical stripe features, respectively. Specifically, the input of the SiSA module $x \in \mathbb{R}^{H \times W \times C}$ is first divided into two groups $x_H \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ and $x_V \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ along the channel dimension. The two groups are then processed by horizontal and vertical branches separately, in which feature maps are treated as a sequence of the horizontal stripes and vertical stripes, respectively. The details of SiSA are shown in Fig. 3 and will be described in Section 3.4. We denote the horizontal and vertical branches of the SiSA module as $SiSA_H$ and $SiSA_V$, respectively. The outputs of these two branches $h_H \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ and $h_V \in \mathbb{R}^{H \times W \times \frac{C}{2}}$ can be obtained by

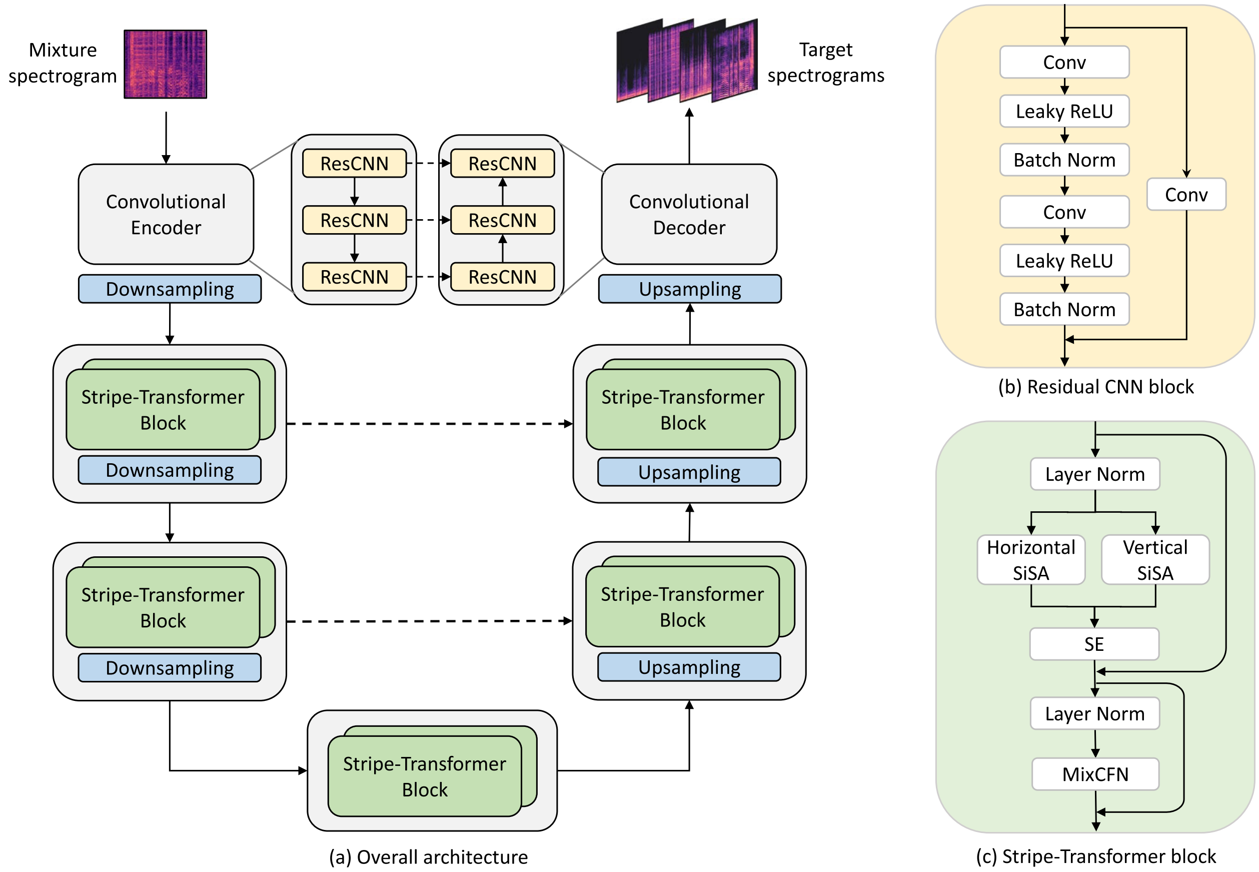


Fig. 2 The overall architecture of the proposed model and the detail of residual CNN and stripe-Transformer block

$$h_H = SiSA_H(LayerNorm(x_H)), \tag{1}$$

$$h_V = SiSA_V(LayerNorm(x_V)). \tag{2}$$

h_H and h_V are then passed into the squeeze-and-excitation (SE) [54] module for feature aggregation along the channel dimension. SE module consists of two linear layers and a sigmoid activation function, which can further enhance important feature maps. The output of the SE module $h \in \mathbb{R}^{H\times W\times C}$ can be obtained by

$$h = x + SE(h_H, h_V). \tag{3}$$

We use mixed-scale convolutional FFN (MixCFN) first proposed in [55] to further process the feature outputs from attention layers. MixCFN is based on the structure of the common feed-forward-network (FFN), which consists of two fully connected (FC) layers and a GELU activation function. To further extract multi-scale local information, MixCFN adds two depth-wise convolution paths between two FC layers. Specifically, the feature maps after the first FC layer are split into two parts along the channel dimension and then passed into 3 × 3 and 5 × 5 depth-wise convolution layers.

Finally, the output of the MixCFN $z \in \mathbb{R}^{H\times W\times C}$, which is also the output of the stripe-Transformer block, can be obtained by

$$z = h + MixCFN(LayerNorm(h)). \tag{4}$$

The layer normalization is used to speed up network convergence and residual connection is used to avoid vanishing gradient problems.

3.4 Stripe-wise self-attention

The SiSA module contains horizontal and vertical branches, as mentioned in the above section. We take the vertical branch of the SiSA module as an example for further explanation, as shown in Fig. 3. The horizontal branch is in a similar pattern to the vertical branch and will not be discussed.

Basically, each feature map in the vertical branch can be modeled as a sequence of vertical stripes, in which each stripe is also a sequence of frequency bins at a certain time

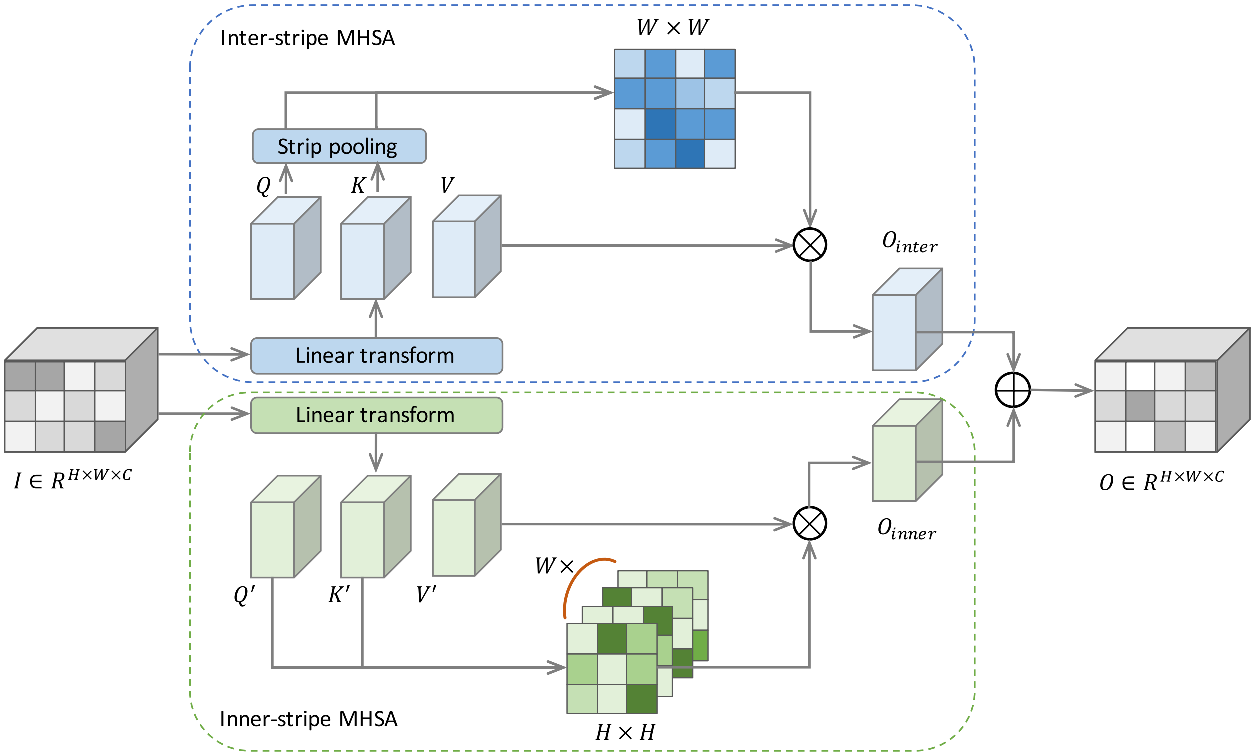


Fig. 3 The illustration of the vertical branch of the stripe-wise self-attention module (*SiSA*$_V$)

step. And it has been demonstrated in previous works [27, 30] that capturing long-term dependencies of feature sequences is beneficial to the source separation task. Therefore, we propose to use two kinds of attention-based networks, inter-stripe multi-head self-attention (MHSA) and inner-stripe MHSA, to deal with the long-term dependency problem of stripe-level feature modeling. The former is used to capture dependencies between different stripes, and the latter is used to capture dependencies in each stripe.

Let $I \in \mathbb{R}^{H \times W \times C}$ be the input feature maps of the vertical SiSA, in which C is half the channel number of the input of the stripe-Transformer block. For inter-stripe MHSA, I is first processed by a linear transform along the channel dimension and reshaped to Q, K, and V $\in \mathbb{R}^{n \times H \times W \times c}$, in which n denotes the head numbers and c denotes the number of channels per head. The strip-pooling [56] strategy is used to down-sample feature maps Q and K as stripe tokens. It performs global average pooling (GAP) on each vertical stripe, which can be denoted as

$$y_j = \frac{1}{H} \sum_{0 \leq i \leq H} x_{i,j}, \tag{5}$$

in which y_j denotes the down-sampled feature of a single stripe. Through this spatial reduction operation, the shape of Q and K becomes $n \times W \times c$. Then, the attention maps $A \in \mathbb{R}^{n \times W \times W}$ can be obtained from the inner product of Q and K, in which an attention map $A_{n_o} \in \mathbb{R}^{W \times W}$ of head n_o can be obtained by

$$A_{n_o} = softmax(\frac{Q_{n_o} \cdot K_{n_o}^T}{\sqrt{c}}). \tag{6}$$

Then, we take the inner product of A and V to obtain the hidden result $Y \in \mathbb{R}^{n \times H \times W \times c}$. Finally, we concatenate the n heads of Y and use one linear transform layer to obtain $O_{\text{inter}} \in \mathbb{R}^{H \times W \times C}$ as the result of inter-stripe MHSA. The obtained attention map A reflects the global dependencies of different stripes in the spectrogram, which will be further analyzed in Section 4.6.

For the inner-stripe MHSA part, matrices $Q^{'}$, $K^{'}$, and $V^{'}$ are similarly obtained from another linear transform layer and reshape operation. While the spatial-reduction operation is not required, the self-attention operation is done inside each vertical stripe separately. The attention maps $A^{'} \in \mathbb{R}^{n \times W \times H \times H}$ can be obtained from the inner product of $Q^{'}$ and $K^{'}$, in which an attention map $A^{'}_{n^{'}_o, w_o} \in \mathbb{R}^{H \times H}$ of a specific head $n^{'}_o$ and stripe w_o can be obtained from

$$A'_{n'_o,w_o} = softmax\left(\frac{Q'_{n'_o,w_o} \cdot K'^{T}_{n'_o,w_o}}{\sqrt{c}}\right). \quad (7)$$

We use the inner product of A' and V' to obtain the hidden result $Y' \in \mathbb{R}^{n \times H \times W \times c}$. Then, we obtain the result of inner-stripe MHSA $O_{\text{inner}} \in \mathbb{R}^{H \times W \times C}$ using the same way as the inter-stripe MHSA mentioned above.

Finally, the output feature maps of the vertical SiSA module $O \in \mathbb{R}^{H \times W \times C}$ can be obtained from the summation of O_{inter} and O_{inner}.

4 Experiments

In this section, we first introduce our experimental settings, including dataset, model configuration, training, and evaluation strategies. And we explore the performance of the proposed model compared with other DNN-based networks. We also perform comparison experiments concerning the construction of the proposed model and the effect of input audio length. Finally, the visualization results and analyses are discussed.

4.1 Dataset

We use the open dataset Musdb18 [57], a professional multi-track dataset that contains four target sources, including "vocals," "bass," "drums," and "other," among which the "other" includes musical instruments except for the previous three, such as piano and violin. The dataset contains 150 pieces in total, including 100 songs in the training set and 50 songs in the testing set. We then divide the songs in the training set into 86 songs for model training and 14 songs for model validation. All audio materials are in stereo format at 44.1kHz.

4.2 Experimental settings

During data generation, random segments in training songs are selected for each training iteration. Each song segment is processed by STFT to obtain the spectrogram, with a window length and hop size of 4096 and 1024 samples, respectively. The size of the obtained spectrogram is 2 × 2049 × 256, which represents a roughly 6-s song piece. Since the frequency bandwidth of the tracks in the Musdb18 dataset is limited to 16kHz, we cut the high-frequency part and finally obtain the spectrogram with the size of 2 × 1536 × 256, which is then fed into the neural networks. We also use data augmentation to obtain more sufficient data for training, mainly the random remixing and random amplitude scaling [58]. The random remixing strategy randomly takes 6-s pieces from each audio source track and then obtains the mixture by the linear summation of all tracks.

For detail settings of our proposed model, the channel numbers of the ResCNN part in the encoder are 32, 48, and 64, and the kernel size keeps 3 × 3. In multi-scale stripe-Transformer blocks, the channel numbers are 128, 256, and 512, and the head numbers are 4, 8, and 16. The expansion ratio for MixCFN keeps 3.

We use the method that decouples the estimation of magnitudes and phases to optimize the model [29], in which complex ideal ratio masks (cIRMs) are obtained from the final layer of the network. The reconstructed source signals are obtained from the product of the input complex spectrogram and the output cIRMs in the complex domain. The loss we use is the L1 loss between reconstructed waveform source signals and ground truths. We use the Adam optimizer without regularization. The learning rate is set to 0.0001 initially and is multiplied by a factor of 0.9 after every 10K steps. The batch size is set to 16. Stripe-Transformer and other comparison models are trained for 200K iterations with four V100 32G GPUs.

For the inference of the model, we refer to the practice of Sams-Net [30] to cut the original complete audio into several continuous segments, and each piece will be fed into the stripe-Transformer network. Finally, the results of all segments are assembled to obtain recovered songs.

We evaluate the proposed model and comparison models by using three objective indicators, namely source-to-distortion ratio (SDR), source-to-interference ratio (SIR), and source-to-artifact ratio (SAR). Given an estimate of a source s_i composed of the true source s_{target}, and three error terms, interference e_{interf}, noise e_{noise}, and artifacts e_{artif} [59], the SDR, SIR, and SAR can be defined as follows:

$$\text{SDR} = 10\log_{10}\left(\frac{\left\|s_{\text{target}}\right\|^2}{\left\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\right\|^2}\right), \quad (8)$$

$$\text{SIR} = 10\log_{10}\left(\frac{\left\|s_{\text{target}}\right\|^2}{\left\|e_{\text{interf}}\right\|^2}\right), \quad (9)$$

$$\text{SAR} = 10\log_{10}\left(\frac{\left\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\right\|^2}{\left\|e_{\text{artif}}\right\|^2}\right). \quad (10)$$

All metrics mentioned above are calculated by the Python package *museval* [60] using the median of frames and median of tracks.

4.3 Ablation study

We design some ablation experiments to verify the effectiveness of stripe-Transformer. Firstly, we verify the performance of the stacked stripe-Transformer blocks as the bottleneck part of the network, compared with stacked blocks using the other two backbone networks. We denote our proposed model as "Stripe-T." The one baseline model we refer to is the residual CNN block, using the structure consistent with the encoder as shown in Fig. 2b. We denote this method as "ResCNNs." The other baseline model we refer to is the spatial-reduction Transformer (SR-T) block in Pyramid Vision Transformer (PVT) [61]. The SR-T block is another Transformer-based network component widely used in image segmentation, which can also use a self-attention mechanism to process high-resolution feature maps. Specifically, it adopts depth-wise CNN with a certain stride before self-attention operation. This spatial-reduction design can avoid high computation costs when capturing global dependencies. In our experiments, the down-sampling stride of the spatial-reduction convolution kernel is set to 8 × 8, 4 × 4, and 2 × 2 in three stages, and the head numbers and channel numbers are the same as Stripe-T. We denote this method as "SR-T." We keep the structure of the encoder and decoder the same as in Fig. 2 and use the same number of stacked three comparison blocks as the bottleneck parts. The numbers of parameters of ResCNNs, SR-T, and Stripe-T are 20.48M, 23.81M, and 10.60M, respectively.

The experimental results can be seen in Table 1. We compare the SDR, SIR, and SAR performance of the mentioned three bottlenecks. According to the SDR value, ResCNNs and SR-T have similar performance, among which ResCNNs is slightly higher than SR-T on "bass" and "other" and slightly lower than SR-T on "drums" and "vocals." Stripe-Transformer outperforms the other two methods on all four targets. According to the average values of the three indicators, ResCNNs and SR-T are both 7.75 dB, while stripe-Transformer reaches 8.52dB, with about 0.77dB improvement compared with the mentioned two methods.

To further explore the construction of the proposed stripe-Transformer, we test the performance of the system when removing some components of the stripe-Transformer, as presented in Table 2. We first verify the effectiveness of the horizontal and vertical branches of the SiSA module. When removing horizontal and vertical branches of stripe-Transformer blocks, the mean of metrics decreases by 0.43dB and 0.30dB, respectively, indicating that the removal of horizontal SiSA has a slightly more significant impact on the performance. We also verify the effect of inner-stripe and inter-stripe MHSA inside the SiSA module. When removing these two parts separately, the mean of metrics decreases by 0.61dB and 0.63dB. In summary, the removal of any branch of the SiSA module will degrade the performance of the proposed system.

4.4 Context length of stripe-Transformer

For Transformer-based networks, the length of the input sequence will affect the performance of the model [30, 62]. We test the performance of the model with different input segment lengths. The metric we use is the average of SDR, SIR, and SAR scores. We set the frame lengths of the input audio to be 64, 128, 256, 512, and 1024, in which 256-frame stands for around 6s in our experimental settings. The results are shown in Fig. 4. It can be found that when the frame length is set to 256, the average score reaches the highest for "vocals" and "drums" separation. The score of "bass" separation achieves the

Table 1 SDR, SIR, and SAR value of U-Net-based models using different backbone networks, evaluated on the test set of Musdb18

	SDR				SIR				SAR				
	Bass	Drums	Other	Vocals	Bass	Drums	Other	Vocals	Bass	Drums	Other	Vocals	Mean
ResCNNs	4.99	6.41	5.23	6.74	9.65	13.19	7.77	15.51	5.88	6.22	4.63	6.73	7.75
SR-T	4.81	6.57	5.09	6.85	10.28	12.73	8.09	14.62	**5.92**	6.47	4.55	6.98	7.75
Stripe-T	**5.46**	**7.56**	**5.82**	**7.75**	**11.00**	**13.36**	**9.00**	**16.37**	5.92	**7.23**	**5.18**	**7.57**	**8.52**

The best metrics are highlighted using bold font

Table 2 Comparison of the experimental configuration of removing different components of the stripe-Transformer

	SDR				SIR				SAR				
	Bass	Drums	Other	Vocals	Bass	Drums	Other	Vocals	Bass	Drums	Other	Vocals	Mean
Stripe-T	**5.46**	**7.56**	**5.82**	**7.75**	11.00	13.36	**9.00**	**16.37**	**5.92**	**7.23**	**5.18**	**7.57**	**8.52**
- Hor. SiSA	4.94	6.98	5.31	7.28	**11.10**	12.90	8.18	15.80	5.78	6.76	4.80	7.24	8.09
- Ver. SiSA	5.17	6.91	5.38	7.27	11.02	**13.66**	8.30	16.00	5.68	6.95	4.95	7.31	8.22
- Inner. MHSA	5.09	6.59	5.27	7.09	10.62	12.64	8.06	15.33	5.53	6.66	4.91	7.12	7.91
- Inter. MHSA	4.90	6.97	5.09	7.07	10.55	12.88	7.85	15.49	5.40	6.55	5.00	6.89	7.89

The best metrics are highlighted using bold font

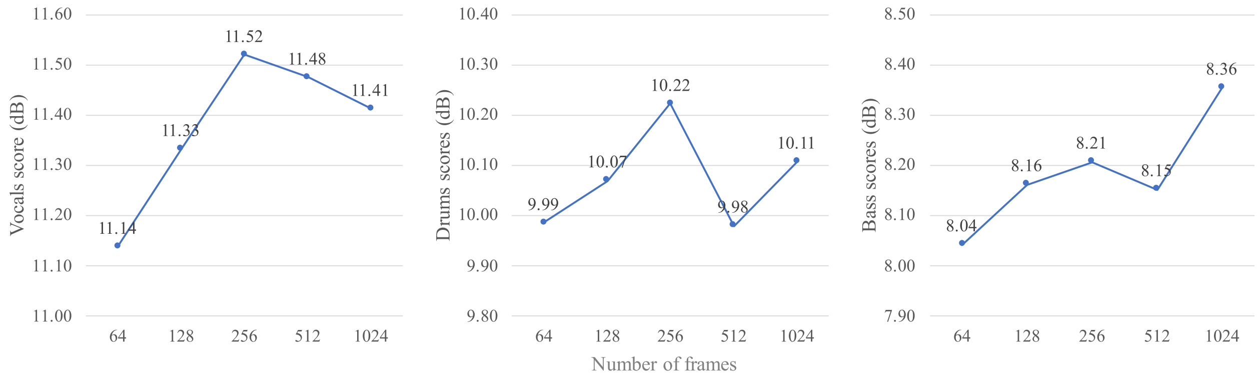


Fig. 4 The average score of SDR, SIR, and SAR on "vocals," "drums," and "bass" categories with different input audio lengths

Table 3 Comparison of the SDR metric with other models on Musdb18

	Vocals	Drums	Bass	Other	Avg.
Wave-U-Net [15]	3.25	4.22	3.21	2.25	3.23
Meta-TasNet [18]	6.40	5.91	5.58	4.19	5.52
Open-Unmix [63]	6.32	5.73	5.23	4.02	5.33
MMDenseLSTM [27]	6.60	6.41	5.16	4.15	5.58
Sams-Net [30]	6.61	6.63	5.25	4.09	5.65
D3Net [31]	7.24	7.01	5.25	4.53	6.01
Deep ResUNet [29]	**8.98**	6.62	**6.04**	5.29	**6.73**
Stripe-Transformer	7.83	**7.63**	5.50	**5.89**	6.71

The best metrics are highlighted using bold font

highest when the number of input frames is 1024. And the separation performance of the above three instruments remains poor when the number of frames is 64, which contains the least context information.

4.5 Comparison with other MSS systems

Table 3 shows the comparison results of SDR score between our model and existing MSS methods. Wave-U-Net [15] is an adaption architecture of U-Net on time-domain representations. Meta-TasNet [18] is a meta-learning-inspired architecture for source separation. The above two methods are all time-domain methods, and the rest are spectrogram-based methods. Open-Unmix [63] is a frequently used benchmark system on the Musdb18 dataset, with three bidirectional LSTMs as the backbone network. MMDenseLSTM [27] uses multi-band multi-scale CNNs [41] and integrates long short-term memory (LSTM). Sams-Net [30] introduces the sliced attention mechanism to the spectrogram domain. D3Net [31] uses densely multi-dilated convolution to further improve the performance based on multi-band configuration. Deep ResUNet [29] uses a 143-layer network with a novel cIRM estimation strategy, which achieves the SOTA performance in spectrogram-based methods.

As shown in Table 3, the proposed stripe-Transformer achieves 7.63dB on the "drums" category and 5.89dB on the "other" category, which outperforms the above systems. On the "vocals" category, the stripe-Transformer achieves 7.83dB, with a significant improvement compared to other methods other than Deep ResUNet. On the "bass" category, stripe-Transformer is comparable with most spectrogram-based methods and is relatively weaker than Deep ResUNet. For the overall performance, stripe-Transformer achieves averagely 6.71dB, which is comparable with the 6.73dB of Deep ResUNet while using around one-tenth of the number of parameters. We compare stripe-Transformer with mentioned methods in terms of the overall performance and model parameters, as summarized in Fig. 5.

4.6 Visualization

To further investigate the effect of the stripe-wise self-attention mechanism of the proposed model, we extract the attention maps of stripe-Transformer. The stripe-attention maps of the vertical branch are taken from the first stage of the bottleneck part, and those of the horizontal branch are taken from the third stage. The results are shown in Fig. 6. The attention score is an average of all query stripes and attention heads, which provides a more global interpretation.

The bottom attention bar profiles vertical stripes of spectrograms, i.e., time steps. It can be found that the most highlighted areas along the time axis are located around the drum signals, especially the time steps of the kick drums. And the left attention bar profiles horizontal stripes of spectrograms, i.e., frequency bands. It can be found that the relatively lower frequency band (<4000 Hz) achieves the higher attention score in the above two

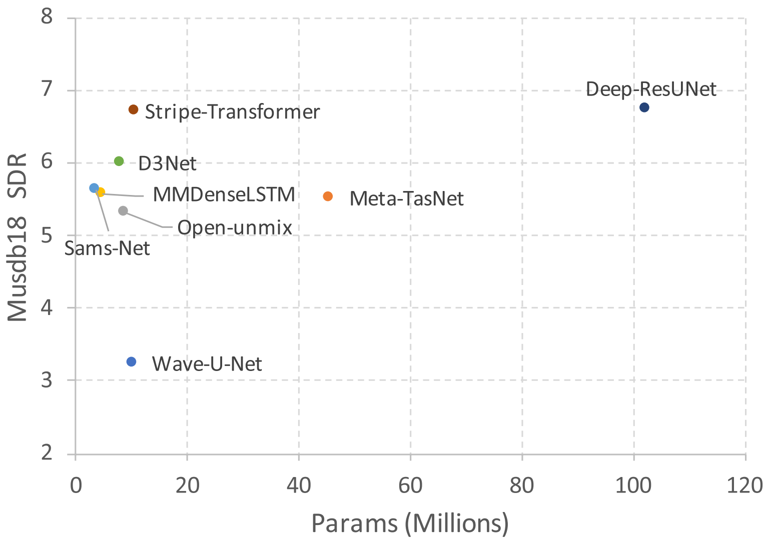
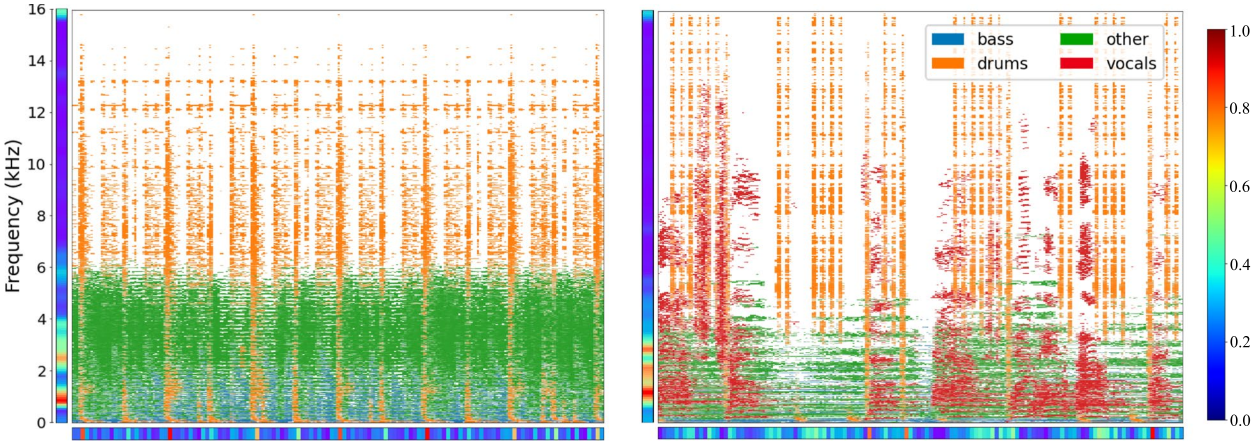


Fig. 5 Performance vs. model parameters on the Musdb18 dataset. Fewer parameters and higher SDR score are better



Girls Under Glass - We Feel Alright

The Mountaineering Club - Mallory

Fig. 6 Visualization of deep stripe feature learning results. Two song clips are taken from the Musdb18 test set. Magnitude spectrograms of the mixture are presented, in which four different colors represent the four constituent sources. On the left and bottom sides of each spectrogram, there are score bars which represent the average attention scores of horizontal and vertical stripes, respectively. The higher the score is, the greater attention the network pays to the position

cases, since the lower frequency band is considered as the most complicated part along the frequency axis, which contains more instrumental energies.

We also evaluate the quality of the reconstructed spectrograms for each target source, using different comparison models mentioned in Section 4.3. As shown in Fig. 7, we use red boxes to highlight differences in target spectrograms estimated by ResCNNs, SR-T, and stripe-Transformer. In terms of the reconstruction outputs of the “drums” category, the vertical stripes are broken while using ResCNNs; the boundary between the vertical stripes is not clear enough using SR-T. In the lower frequency part, the constituents of the music are often more complicated, which makes the separation of drum activities more likely to make mistakes. In comparison, stripe-Transformer can better recover these details. Since drum activities are shown as vertical stripes in the spectrogram, their locations and relationships can be better handled by stripe-level feature modeling. For the “bass” category, most of the energy in the red box is lost when using ResCNNs while it is preserved well when using SR-T and stripe-Transformer. It demonstrates the importance of capturing global dependencies using some strategies such as

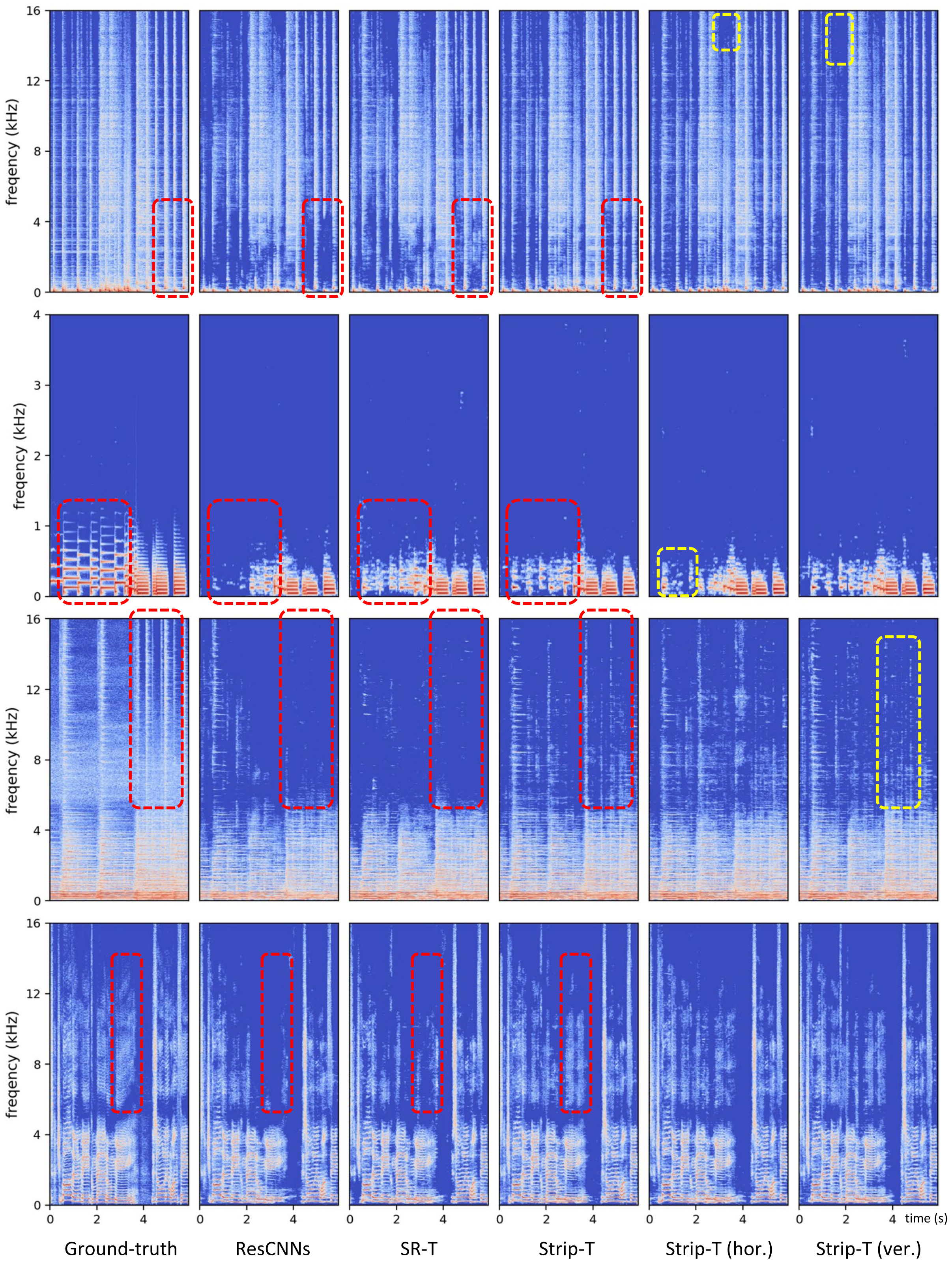


Fig. 7 Comparison of reconstructed magnitudes using different backbone networks. The four rows from top to bottom are drums, bass, other, and vocals target source, which is taken from a roughly 6-s piece of *Schoolboy Fascination*

the self-attention mechanism. For the “other” category, ResCNNs and SR-T drop percussive signals in the red box, which appear as vertical stripes in the spectrogram. For the “vocals” category, stripe-Transformer can also better recover the labeled region compared with ResCNNs and SR-T, with its better ability to process harmonic signals shown as horizontal stripes.

And we use yellow boxes to highlight differences made by removing horizontal or vertical SiSAs in stripe-Transformer. For the “drums” category, both two models miss high-frequency energies at labeled regions. For the “bass” category, the stripe-Transformer with only horizontal SiSAs almost misses one note. For the “other” category, there are no significant percussive signals in the labeled spectrogram estimated by the stripe-Transformer with only vertical SiSAs. Therefore, it can be demonstrated that the removal of the horizontal or vertical branch of the SiSA module might degrade the performance.

5 Conclusion

In this paper, we propose a novel deep neural network architecture, stripe-Transformer, for the task of music source separation. The stripe feature learning module in the proposed model significantly boosts the performance of MSS. The experimental results on the Musdb18 dataset show that the proposed model achieves SOTA performance with fewer parameters in terms of SDR score. The quality of reconstructed spectrograms is better when using stripe-Transformer compared with ResUNet and SR-T. And visualization results of attention maps show that our proposed model can better highlight beat and harmonic structures in music spectrograms.

In our future work, we will enlarge the proposed network and apply it into more instrumental separation tasks. Moreover, Transformer-based networks usually need large amount of training data. We will further investigate data augmentation techniques and some semi-supervised methods such as noisy self-training [64] to further improve the performance of the model.

Abbreviations

MSS	Music source separation
MIR	Music information retrieval
SDR	Source-to-distortion ratio
DNN	Deep neural network
CNNs	Convolutional neural networks
RNNs	Recurrent neural networks
SIMO	Single-input-multi-output
SiSA	Stripe-wise self-attention
SE	Squeeze-and-excitation
MixCFN	Mixed-scale convolutional FFN
MHSA	Multi-head self-attention
GAP	Global average pooling
STFT	Short-time Fourier transform
cIRMs	Complex ideal ratio masks
SIR	Source-to-interference ratio
SAR	Source-to-artifact ratio
Stripe-T	Stripe-Transformer
ResCNNs	Residual convolutional neural networks
SR-T	Spatial-reduction Transformer
PVT	Pyramid Vision Transformer
LSTM	Long short-term memory
ResUNet	Residual U-Net

Acknowledgements
Not applicable.

Authors’ contributions
JQ designed the system, implemented the experiments, and wrote the paper. XL helped implement experiments, designed some of the figures, and wrote the corresponding descriptions. YY and WL edited the paper and provided the paper correction. The authors read and approved the final manuscript.

Funding
This work was supported by the National Key R &D Program of China (2019YFC1711800) and NSFC (62171138).

Availability of data and materials
The Musdb18 dataset used during the current study are available at https://sigsep.github.io/datasets/musdb.html.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 3 September 2022 Accepted: 21 December 2022
Published online: 12 January 2023

References

1. J. Pons, J. Janer, T. Rode, W. Nogueira, Remixing music using source separation algorithms to improve the musical experience of cochlear implant users. J. Acoust. Soc. Am. **140**(6), 4338–4349 (2016)
2. A.J. Simpson, G. Roma, M.D. in *International Conference on Latent Variable Analysis and Signal Separation*. Plumbley, Deep karaoke: extracting vocals from musical mixtures using a convolutional deep neural network (Springer, 2015), pp. 429–436
3. A. Rosner, B. Kostek, Automatic music genre classification based on musical instrument track separation. J. Intell. Inf. Syst. **50**(2), 363–384 (2018)
4. A. Rosner, B. Kostek, in *International Symposium on Methodologies for Intelligent Systems*. Musical instrument separation applied to music genre classification (Springer, 2015), pp. 420–430
5. J.S. Gómez, J. Abeßer, E. Cano, in *ISMIR*. Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning (ISMIR, Paris, 2018), pp. 577–584
6. B. Sharma, R.K. Das, H. Li, in *INTERSPEECH*. On the importance of audio-source separation for singer identification in polyphonic music (ISCA, Graz, 2019), pp. 2020–2024
7. Y. Gao, X. Zhang, W. Li, Vocal melody extraction via HRNnet-based singing voice separation and encoder-decoder-based F0 estimation. Electronics **10**(3), 298 (2021)
8. J. Xu, X. Li, Y. Hao, G. Yang, in *Proceedings of international conference on multimedia retrieval*. Source separation improves music emotion recognition (ACM, Glasgow, 2014), pp. 423–426
9. E. Alfaro-Paredes, L. Alfaro-Carrasco, W. Ugarte, in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Query by humming for song identification using voice isolation (Springer, 2021), pp. 323–334
10. R. Kumar, Y. Luo, N. Mesgarani, in *Interspeech*. Music source activity detection and separation using deep attractor network (ISCA, Hyderabad, 2018), pp. 347–351

11. D. Stoller, S. Ewert, S. Dixon, in *International Conference on Latent Variable Analysis and Signal Separation*. Jointly detecting and separating singing voice: a multi-task approach (Springer, 2018), pp. 329–339
12. Y. Hung, A. Lerch, in *ISMIR*. Multitask learning for instrument activation aware music source separation (ISMIR, Montréal, 2020), pp. 748–755
13. T. Nakano, K. Yoshii, Y. Wu, R. Nishikimi, K.W.E. Lin, M. Goto, in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. Joint singing pitch estimation and voice separation based on a neural harmonic structure renderer (IEEE, 2019), pp. 160–164
14. A. Jansson, R.M. Bittner, S. Ewert, T. Weyde, in *2019 27th European Signal Processing Conference (EUSIPCO)*. Joint singing voice separation and f0 estimation with deep u-net architectures (IEEE, 2019), pp. 1–5
15. D. Stoller, S. Ewert, S. Dixon, in *ISMIR*. Wave-u-net: a multi-scale neural network for end-to-end audio source separation (ISMIR, Paris, 2018), pp. 334–340
16. F. Lluís, J. Pons, X. Serra, in *INTERSPEECH*. End-to-end music source separation: is it possible in the waveform domain? (ISCA, Graz, 2019), pp. 4619–4623
17. A. Défossez, N. Usunier, L. Bottou, F. Bach, Music source separation in the waveform domain. arXiv preprint arXiv:1911.13254 (2019)
18. D. Samuel, A. Ganeshan, J. Naradowsky, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Meta-learning extractors for music source separation (IEEE, 2020), pp. 816–820
19. B. Zhu, W. Li, R. Li, X. Xue, Multi-stage non-negative matrix factorization for monaural singing voice separation. IEEE Trans. Audio Speech Lang. Process. **21**(10), 2096–2107 (2013)
20. B. Rathnayake, K. Weerakoon, G. Godaliyadda, M. Ekanayake, in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. Toward finding optimal source dictionaries for single channel music source separation using nonnegative matrix factorization (IEEE, 2018), pp. 1493–1500
21. T. Virtanen, A. Mesaros, M. Ryynänen, in *SAPA@ INTERSPEECH*. Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music (ISCA, Brisbane, 2008), pp. 17–22
22. X. Zhang, W. Li, B. Zhu, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Latent time-frequency component analysis: a novel pitch-based approach for singing voice separation (IEEE, 2015), pp. 131–135
23. C.L. Hsu, D. Wang, J.S.R. Jang, K. Hu, A tandem algorithm for singing pitch extraction and voice separation from music accompaniment. IEEE Trans. Audio Speech Lang. Process. **20**(5), 1482–1491 (2012)
24. Z. Rafii, B. Pardo, in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. A simple music/voice separation method based on the extraction of the repeating musical structure (IEEE, 2011), pp. 221–224
25. Y. Zhang, X. Ma, in *International Symposium on Neural Networks*. A singing voice/music separation method based on non-negative tensor factorization and repeat pattern extraction (Springer, 2015), pp. 287–296
26. Z. Rafii, B. Pardo, Repeating pattern extraction technique (REPET): a simple method for music/voice separation. IEEE Trans. Audio Speech Lang. Process **21**(1), 73–84 (2012)
27. N. Takahashi, N. Goswami, Y. Mitsufuji, in *2018 16th International workshop on acoustic signal enhancement (IWAENC)*. Mmdenselstm: an efficient combination of convolutional and recurrent neural networks for audio source separation (IEEE, 2018), pp. 106–110
28. A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, T. Weyde, in *ISMIR*. Singing voice separation with deep u-net convolutional networks (ISMIR, Suzhou, 2017), pp. 745–751
29. Q. Kong, Y. Cao, H. Liu, K. Choi, Y. Wang, in *ISMIR*. Decoupling magnitude and phase estimation with deep resunet for music source separation (ISMIR, 2021), pp. 342–349
30. T. Li, J. Chen, H. Hou, M. Li, in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. Sams-net: a sliced attention-based neural network for music source separation (IEEE, 2021), pp. 1–5
31. N. Takahashi, Y. Mitsufuji, D3net: densely connected multidilated densenet for music source separation. arXiv preprint arXiv:2010.01733 (2020)
32. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. **25**, pp. 1097–1105 (2012)
33. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Deep residual learning for image recognition (IEEE, Las Vegas, 2016), pp. 770–778
34. G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Densely connected convolutional networks (IEEE, Honolulu, 2017), pp. 4700–4708
35. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput **9**(8), 1735–1780 (1997)
36. K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, in *EMNLP*. Learning phrase representations using RNN encoder-decoder for statistical machine translation (ACL, Doha, 2014), pp. 1724–1734
37. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Adv Neural Inf Process Syst. **30**, pp. 5998–6008 (2017)
38. J. Devlin, M. Chang, K. Lee, K. Toutanova, in *NAACL-HLT*. BERT: pre-training of deep bidirectional transformers for language understanding (1) (ACL, Minneapolis, 2019), pp. 4171–4186
39. R.P. Bingham, Harmonics-understanding the facts. Dranetz Technol. (Citeseer, Edison, 1994)
40. P. Chandna, M. Miron, J. Janer, E. Gómez, in *International conference on latent variable analysis and signal separation*. Monoaural audio source separation using deep convolutional neural networks (Springer, 2017), pp. 258–266
41. N. Takahashi, Y. Mitsufuji, in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. Multi-scale multi-band densenets for audio source separation (IEEE, 2017), pp. 21–25
42. S. Park, T. Kim, K. Lee, N. Kwak, in *ISMIR*. Music source separation using stacked hourglass networks (ISMIR, Paris, 2018), pp. 289–296
43. O. Ronneberger, P. Fischer, T. Brox, in *International Conference on Medical image computing and computer-assisted intervention*. U-net: convolutional networks for biomedical image segmentation (Springer, 2015), pp. 234–241
44. V.S. Kadandale, J.F. Montesinos, G. Haro, E. Gómez, in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. Multi-channel u-net for music source separation (IEEE, 2020), pp. 1–6
45. R. Hennequin, A. Khlif, F. Voituret, M. Moussallam, Spleeter: a fast and efficient music source separation tool with pre-trained models. J Open Source Softw. **5**(50), 2154 (2020)
46. W. Choi, M. Kim, J. Chung, D. Lee, S. Jung, Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation. arXiv preprint arXiv:1912.02591 (2019)
47. J. Perez-Lapillo, O. Galkin, T. Weyde, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Improving singing voice separation with the Wave-U-Net using minimum hyperspherical energy (IEEE, 2020), pp. 3272–3276
48. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
49. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Swin transformer: hierarchical vision transformer using shifted windows (IEEE, Piscataway, 2021), pp. 10012–10022
50. L. Dong, S. Xu, B. Xu, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition (IEEE, 2018), pp. 5884–5888
51. N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, in *Proceedings of the AAAI Conference on Artificial Intelligence*. Neural speech synthesis with transformer network, vol. 33 (AAAI, Honolulu, 2019), pp. 6706–6713
52. S. Yu, C. Li, F. Deng, X. Wang, in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Rethinking singing voice separation with spectral-temporal transformer (IEEE, 2021), pp. 884–889
53. Y. Luo, Z. Chen, C. Han, C. Li, T. Zhou, N. Mesgarani, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rethinking the separation layers in speech separation networks (IEEE, 2021), pp. 1–5

54. J. Hu, L. Shen, G. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Squeeze-and-excitation networks (IEEE, Salt Lake City, 2018), pp. 7132–7141
55. J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.H. Chen, L. Lai, V. Chandra, D.Z. Pan, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Multi-scale high-resolution vision transformer for semantic segmentation (IEEE, New Orleans, 2022), pp. 12094–12103
56. Q. Hou, L. Zhang, M.M. Cheng, J. Feng, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Strip pooling: rethinking spatial pooling for scene parsing (IEEE, Piscataway, 2020), pp. 4003–4012
57. Z. Rafii, A. Liutkus, F.R. Stöter, S.I. Mimilakis, R. Bittner, The MUSDB18 corpus for music separation (2017). https://doi.org/10.5281/zenodo.1117372.
58. S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, Y. Mitsufuji, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Improving music source separation based on deep neural networks through data augmentation and network blending (IEEE, 2017), pp. 261–265
59. E. Vincent, R. Gribonval, C. Févotte, Performance measurement in blind audio source separation. IEEE Trans. Audio Speech Lang. Process. **14**(4), 1462–1469 (2006)
60. F.R. Stöter, A. Liutkus, N. Ito, in *International Conference on Latent Variable Analysis and Signal Separation*. The 2018 signal separation evaluation campaign (Springer, 2018), pp. 293–305
61. W. Wang, E. Xie, X. Li, D.P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions (IEEE, Piscataway, 2021), pp. 568–578
62. T. Hori, N. Moritz, C. Hori, J. Le Roux, in *Interspeech*. Transformer-based long-context end-to-end speech recognition (ISCA, Shanghai, 2020), pp. 5011–5015
63. F.R. Stöter, S. Uhlich, A. Liutkus, Y. Mitsufuji, Open-unmix-a reference implementation for music source separation. J. Open Source Softw. **4**(41), 1667 (2019)
64. Z. Wang, R. Giri, U. Isik, J.M. Valin, A. Krishnaswamy, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Semi-supervised singing voice separation with noisy self-training (IEEE, 2021), pp. 31–35

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.