TF-ATTENTION-NET: AN END TO END NEURAL NETWORK FOR SINGING VOICE SEPARATION

Tingle Li[* 1,2], *Jiawei Chen*[2], *Haowen Hou*[3], *Ming Li*[1]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]School of Computer Science and Technology, Tiangong University, Tianjin, China
[3]AI Platform Department, Tencent Inc., Shenzhen, China
ml442@duke.edu

ABSTRACT

In terms of source separation task, deep neural networks have two major approaches: one approach is modeling in the spectrogram domain, and the other approach is modeling in the waveform domain. Most of the previous papers use CNNs or LSTMs. However, due to the high sampling rate of audio, whether it is LSTMs with long-distance dependent or CNNs with sliding windows, it is still difficult to extract long-term input context. In this case, we propose an end-to-end network: Time-Frequency Attention Net (TF-Attention-Net), to study the ability of the attention mechanism in the source separation task. First, we introduce the Slice Attention, which can extract the acoustic features of temporal and frequency scales under different channels. Besides, the attention mechanism can be parallel calculated, while LSTMs cannot, because of its time-dependent property. Meanwhile, the receptive field of the attention mechanism is larger than the CNNs, which means that we can use shallower layers to extract longer distance dependence. Experiments indicate that our proposed TF-Attention-Net outperforms both the spectrogram-based U-Net and the waveform-based Wave-U-Net baselines.

Index Terms— Source Separation, Music Information Retrieve, Attention Mechanism, Singing Voice Separation

1. INTRODUCTION

In the general source separation problem, we are given several pieces of music that consist of different source signals [1], which may include vocals, drums, bass, or guitar. Our goal is to separate them into two parts: vocals and accompaniments.

Currently, most of the existing successful audio source separation technologies can be categorized into two main types, namely spectrogram based methods [2, 3, 4], and the waveform based methods [5, 6, 7]. In most cases, CNN or LSTM is employed as the feature extractor, but all of them have certain drawbacks. On one hand, for CNN, the receptive field may has some limitations [8]. For example, deeper CNN layers are required to obtain a wider receptive field. Moreover, although the pooling layer expands the receptive field and aggregates the context, it loses the spectrogram details and spatial information. On the other hand for LSTM, it can not perform parallel calculations, which makes the calculation not very efficient. Furthermore, it still can not solve the long-term dependency problem very well [9].

Vaswani et al. proposed a novel neural network structure - Transformer [10], which uses only the attention mechanism structure to obtained the SOTA result in the English-French translation task. Transformer is a structure that can automatically capture sequence distribution. Experiments show that Transformer is more suitable for processing sequences than LSTM, because it can solve long-term dependency problems better than LSTM [10]. In addition, since Transformer has no time-dependent limitation for calculating, it can be computed in parallel. Furthermore, Transformer has a larger receptive field compared to CNN with the same number of layers.

In this paper, we discuss the feature extraction capability of the encoder of Transformer in the source separation task, and propose an end-to-end neural network for singing voice separation named as TF-Attention-Net. It apply sliced attention to both the temporal and frequency axises of each channel. Experimental results show that our model has achieved a new state-of-the-art result.

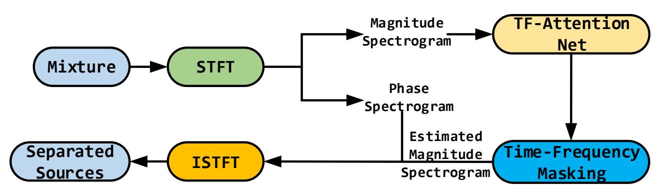

Fig. 1. The flow chart of our training system.

2. SINGING VOICE SEPARATION

Our network designed on the spectrogram as in [11, 12, 13], and the mono music mixture $x \in \mathbb{R}^{1\times T}$ can be expressed as a linear combination of c sources $s \in \mathbb{R}^{1\times T}$:

$$x(t) = \sum_{i=0}^{c} s_i(t) \tag{1}$$

Then we take the Short Fourier Transform (STFT) of each segment, so that each music segment $S_i(t,f)$ can be mapped to a 2D array of time-frequency bins $X(t,f)$:

$$X(t,f) = \sum_{i=0}^{c} S_i(t,f) \tag{2}$$

We assume that the phase of the mix audio is the same as that of the original, so we only take the magnitude of the music segments $|X_i(t,f)|$ as the inputs for the model. The estimated source $\hat{\mathbf{S}}$ is calculated by performing element-wise multiplication between the mask M_i and the mixture spectrogram $|X_i(t,f)|$, then multiply by the imaginary phase power of e.

* Tingle performed this work as an intern at Duke Kunshan University.

$$\widehat{S}_i(t,f) = (|X(t,f)| \otimes M_i(t,f)) \times e^{\varphi(X(t,f))i} \tag{3}$$

where $\varphi(X(t,f))$ is the phase of mixed audio, and M is Wiener-Filter Like Mask (WFM) [14], which can be used to estimate the individual sources. The formula is as follows:

$$M_i(t,f) = \frac{|S_i(t,f)|^2}{\sum_{i=1}^{c} |S_i(t,f)|^2} \quad \text{s.t.} \sum_{i=1}^{c} M_i = 1 \tag{4}$$

So this task can be denoted as minimizing the following objective function:

$$\mathcal{L} = \arg\min_{M} \sum_{i=1}^{c} \left\| S_i(t,f) - \widehat{S}_i(t,f) \right\|_2^2 \tag{5}$$

3. PROPOSAL: TF-ATTENTION-NET

Inspired by the fact that BERT [15] can deal with sequence problems very efficiently in the machine translation task, we proposed the TF-Attention-Net model, which inherit the encoder of the Transformer and reconstructs the mask of each source by focusing on the global features at time-frequency bins simultaneously.

The structure of the model is shown in figure 2. First, we embed it into the k-dimensional space through an CNN layer. For each time-frequency bin, we apply the temporal attention module and the frequency attention module to capture the characteristic dependency of the global feature [16] (we will describe them in detail in Section 3.3). To further enhance the expressive ability of the network, we design a Position-wise CNN (PWC) layer (we will discuss it in Section 3.4) to get vocal and accompaniment information at different resolution scales. Next, we will introduce the details of the TF-Attention-Net.

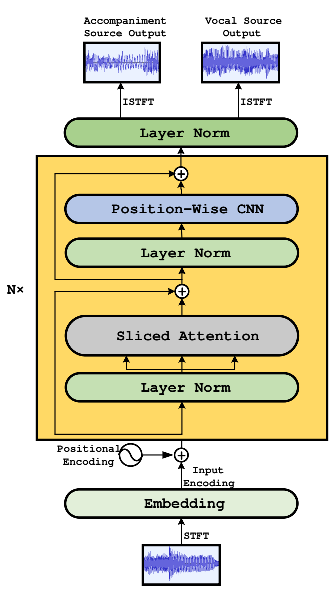

Fig. 2. Model architecture of the TF-Attention-Net.

3.1. Scaled Dot-Product Attention

The attention mechanism is a method of learning the distribution of its own weight by calculating the similarity of sequences. The commonly used methods of attention mechanism are dot product, concatenation, perceptron and etc. Here we use dot product. Specifically, it divides a sequence into three components, query Q , key K , and value V, and then calculates the attention of the three elements. The formula is as follows:

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where $1/\sqrt{d_k}$ plays a regulating role, so that the inner product is not too large and avoid gradient vanishing problem.

3.2. Sliced Attention

In the singing voice separation task, it is better to slice it in advance instead of directly scaling the audio signal to Scaled Dot-Product Attention. Because singing voice separation is not like machine translation, it does not require too long-term dependencies (details are provided in Section 4), but require more useful sample values to be combined to predict the current sample value. So we proposed a new method called Sliced Attention rather than the Scaled Dot-Product Attention stacking like Multi-Head Attention, which slices the input metrics Q, K, V and it can be expressed as:

$$\begin{cases} Q = \{Q_{s_1}, Q_{s_2}, ..., Q_{s_i}\} \\ K = \{K_{s_1}, K_{s_2}, ..., K_{s_i}\} \\ V = \{V_{s_1}, V_{s_2}, ..., V_{s_i}\} \end{cases} \tag{7}$$

Every three $Q_{s_i}, K_{s_i}, V_{s_i}$ is combined as a group $(Q_{s_i}, K_{s_i}, V_{s_i})$. Each group performs the Scaled Dot-Product Attention, then all attention results are concatenated as the output, which can be expressed as:

$$slice_i = Attention(Q_{s_i}, K_{s_i}, V_{s_i}) \tag{8}$$

$$Sliced(Q,K,V) = Concat(slice_1, slice_2, ..., slice_i) \tag{9}$$

3.3. Time-frequency Attention

The general attention usually applies in the 2-dimensional sequence, but spectrogram is a 3-dimensional sequence, i.e., [temporal, frequency, channel]. As pointed out in [6], mapping the number of channels to a higher-dimensional space will result in better separation, so we embed the channel of the spectrogram into the dimension d_k, then do the temporal sliced attention and frequency sliced attention operations. The former is sliced by the time axis while the latter is sliced by the frequency axis. As shown in figure 3, the block on the left operates in the time domain, while the right one operates in the frequency domain. Finally, we stack the results of the two operations in the channel axis and use a CNN to compress the dimensions. The above illustration can be expressed by the following formula ($W^O \in \mathbb{R}^{d_{2k} \times d_k}$ is the convolution operation):

$$Time = Sliced(Q,K,V) \tag{10}$$

$$Freq = Sliced(Q^T, K^T, V^T)^T \tag{11}$$

$$TF\text{-}Attention = Concat(Time, Freq) * W^O \tag{12}$$

3.4. Position-Wise CNN

In addition to the slice attention sub-layers, each block also contains a Position-wise CNN(PWC), which includes three CNN and Batch Normalization layers, where the number of channels after three CNN layers is the same as before the convolution, and the output of the first two Batch Normalization layers is activated by the ReLU function. Meanwhile, the input of this layer is also connected with the

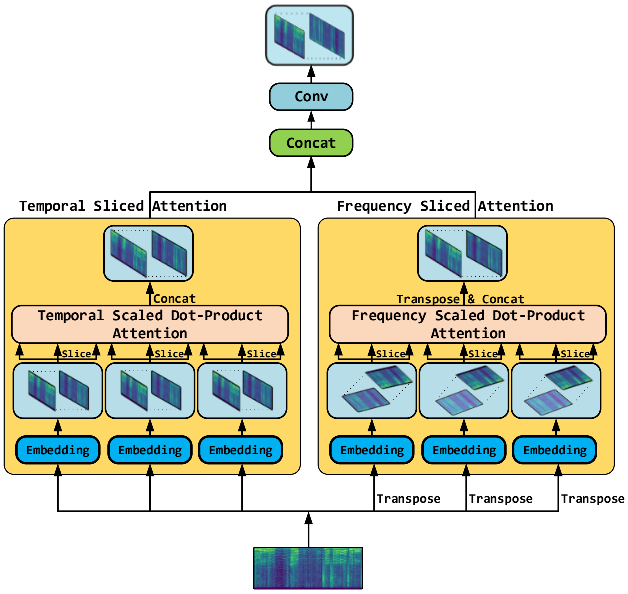

Fig. 3. Model architecture of the Sliced TF Attention Module.

residual after the third convolution, so the output of the PWC can be expressed as:

$$PWC(Z) = \max(0, \max(0, Z*W_1+b_1) \\ *W_2+b_2)*W_3+Z \tag{13}$$

where Z is the output of TF-Attention, $W_i \in \mathbb{R}^{d_k \times d_k}$, $b_i \in \mathbb{R}^{d_k}$ and $Z*W$ is convolution operation.

3.5. Layer Norm

We use residual connections in the two sub-layers above, and then perform layer normalization [17]. So the output of each sublayer is:

$$LayerNorm(x + Sublayer(x)) \tag{14}$$

where $Sublayer(x)$ is a function implemented by the sublayer itself.

3.6. 3D-Positional-Encoding

In general, the magnitude of the spectrogram has a relative time information, but when calculating attention, the dot product is calculated by using each voice sample point and the rest of sample points, which will ignore the order information, that is, no matter how the order of voice is disturbed, the output will get similar results. In order to solve this problem, we can use 3D-Positional-Encoding to get the position information, so that the model can distinguish the spectrogram of different positions:

$$\begin{aligned} PE(\text{pos}, 2i) &= \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \\ PE(\text{pos}, 2i+1) &= \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \end{aligned} \tag{15}$$

where pos = $f(t, f)$, which is a position function about frequency and temporal domain, i denotes the current number of channels.

4. EXPERIMENT AND RESULTS

4.1. Experimental setup

We perform experiments on the MUSDB18 [18] and CCMixter [19] datasets. In order to compare with the baselines, the proportions of training set, validation set and test set we used are consistent with [6], that is, we randomly select 75 tracks from the training set of MUSDB18 and take all the CCMixter dataset as the training set, and we use the remaining 25 tracks of MUSDB18 as the validation set to determine whether it is needed to early stop in case of overfitting. At the end of the training stage, we used all the test sets of MUSDB18, i.e., 50 tracks, to evaluate the model. All the data are converted to mono and downsampling to 8 kHz since [4] first used this to speed up processing.

The TensorFlow framework [20] is used to build our models, and we train them with 1 NVIDIA Tesla M40 GPU. During the training process, we randomly extract the audio clips and fill them correspondingly based on the context of the audio clips in order to align the dimensions, because the length of each song is not the same. The window function of the STFT is set to the Hamming window, and the frame length and frame step are set to 1024 and 768, respectively. We use ADAM optimizer with a learning rate of 0.0001 and the decay rates β_1 = 0.9 and β_2 = 0.999. We set the batch size to 2 and set 2000 steps as one epoch. If the validation loss does not descend after 20 epoch, the early stop will be performed and the training stage will end.

We use signal-to-distortion ratio (SDR) [21] as the evaluation metric, which mainly divides the audio track into multiple non-overlapping audio segments. We also take the method proposed in [6] to evaluate performances in order to make the SDR metric more precisely. Since the distribution of SDR values may not normal, the mean and standard deviation (SD) are not sufficient to describe it. We also employ the median and median absolute deviation (MAD) to describe the performance.

4.2. Results and Discussions

As shown in table 1, we find Slice Attention (slice=2) is better than traditional Scaled Dot-Product Attention(slice=1), when the number of slices increases from 1 to 2, the performance of our model increases. However, when the slices further increase from 2 to 4 and 8, the performance drops gradually. When the number of slices is too large, it will cause the segments to be too small, resulting in the loss of information. The optimal slices number for Slice Attention maybe can be determined by unsupervised clustering methods without knowing the number of clusters on the dataset.

Why Slice Attention is better? Unlike the machine translation task, in the singing voice separation task, the interdependence of each segment is not that stable, because it often has a sudden change in style, such as playing with drums at that moment, but suddenly change to bass or a human voice at the next moment. Under this circumstance, if we focus on the whole sequence, it will attend to irrelevant points, resulting in a lack of focus on the key points.

The Temporal Slice Attention is segmented on the frequency axis, and each slice stores the complete time information. For each instrument, the frequency has a regular harmonic structure on an entire temporal segment, which means the weight of one of the instruments may be larger in a certain frequency band, making the focus of each slice different. For example, one of the slices will focus more on vocals while the others focus more on instruments. Similarly, Frequency Slice Attention is segmented on the temporal axis, and each slice retains complete frequency information. At the same

Table 1. Comparison SDR (in dB) of each model. Best performances among comparison models are shown in bold. (S-Slice, C-Channel)

			Voc.				Acc.			
	# Param	Epoch	Med.	MAD	Mean	SD	Med.	MAD	Mean	SD
1S-128C-3Attention	63.23M	102	4.34	2.91	1.97	10.85	**7.67**	2.01	**7.80**	3.37
2S-128C-3Attention	63.23M	92	**4.34**	2.83	**2.14**	10.41	7.63	2.03	7.77	3.37
4S-128C-3Attention	63.23M	86	4.30	2.85	1.83	10.88	7.57	2.04	7.71	3.40
8S-128C-3Attention	63.23M	101	4.19	2.88	1.19	11.67	7.53	2.05	7.67	3.41
1S-64C-3Attention	15.86M	83	4.15	2.86	1.34	11.56	7.50	2.03	7.67	3.40
2S-64C-3Attention	15.86M	147	**4.19**	2.88	**1.97**	10.42	**7.58**	2.01	**7.74**	3.36
4S-64C-3Attention	15.86M	151	4.23	2.87	1.73	10.87	7.56	2.02	7.71	3.43
4S-64C-3Attention(Temporal)	10.79M	168	4.01	2.80	0.46	12.92	7.46	2.04	7.63	3.36
4S-64C-3Attention(Frequency)	10.79M	101	3.85	2.76	0.89	11.75	7.39	2.01	7.55	3.39
8S-64C-3Attention	15.86M	100	4.04	2.90	1.67	10.59	7.45	2.03	7.63	3.39
4S-64C-2Attention	10.58M	152	4.01	2.79	1.02	11.79	7.47	2.04	7.62	3.35
8S-64C-2Attention	10.58M	76	3.90	2.77	0.54	12.4	7.39	2.02	7.55	3.36
4S-32C-2Attention	2.67M	96	3.74	2.73	0.25	12.74	7.29	2.00	7.45	3.37
8S-32C-2Attention	2.67M	158	3.75	2.69	0.13	12.85	7.30	1.99	7.46	3.36
4S-32C-3Attention	3.99M	127	3.87	2.79	0.73	12.10	7.41	2.03	7.58	3.33
U-Net	224.83M	103	2.90	2.56	-0.42	12.59	6.87	1.99	7.01	3.59
Wave-U-Net	235.69M	72	3.43	2.79	-0.27	13.03	7.11	2.05	7.13	4.01

time, the scope of attention is narrowed down so that the slices are more concerned with the features that are most likely to affect each other's recent time period.

Also, we found that when the number of the channels was increased from 32 to 64 and 128, the separation performance improved and training converges faster at the same time. Besides, we found that the number of attention modules increased from 2 to 3, the separation effect will also be better, but due to the large use of GPU memory, we only trained 3 attention modules. In order to compare with our models, we also trained the baselines: U-Net [1] [4] and Wave-U-Net [1] [6], and the results are shown in table 1, too.

Moreover, compared with the U-Net and Wave-U-Net baselines, the parameters of our model are much smaller. Our model with 100 times less parameters has surpassed the best U-Net results. Further increasing of the parameters can improve the performance. The small number of parameters in our model brings great advantages, such as smaller memory usage and less computation complexity.

We plot the boxplot and the track-by-track comparison picture to compare our model with U-Net [4] and Wave-U-Net [6], which can be seen in figure 4 and 5. In general, the mean of SDR of the vocals and accompaniments in our model is larger than the U-Net and Wave-U-Net. Furthermore, our model achieves better performance than U-Net in 98% vocal and accompaniment tracks. Meanwhile, our model outperforms Wave-U-Net in 92% vocal tracks and in 96% accompaniment tracks. We also provide several online samples [2] from our model, U-Net and Wave-U-Net.

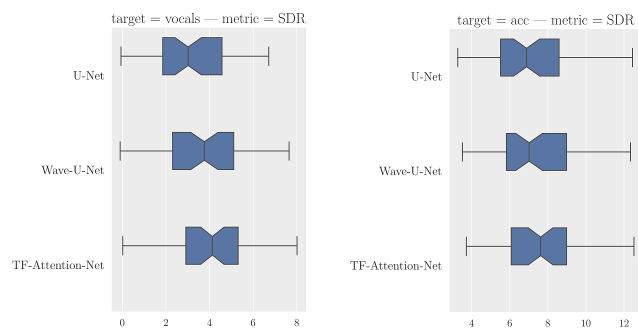

Fig. 4. Details of SDR result for all metrics and methods, where TF-Attention-Net is our model.

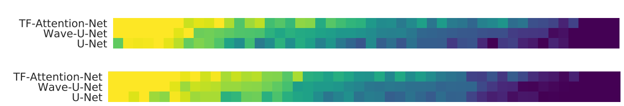

Fig. 5. Track-by-track SDR comparison of the 50 test songs (Vocals (top) and Accompaniment (bottom)). The horizontal axis represents the index of 50 songs, and the color represents the value of SDR. The brighter the color of the block (black-green-yellow), the larger value of the SDR, which means the better the separation performance.

5. CONCLUSIONS

In this paper, we propose an end-to-end spectrum-based audio source separation network called TF-Attention-Net, which apply the attention mechanism to the singing voice separation task. In the evaluation stage, we used a median-based metric that [6] proposed to solve the problem of evaluating abnormalities in quiet audio segments in SDR analysis. As shown by the experiment results, our model achieved better SDR performance although the parameters of which are much less than the U-Net's and Wave-U-Net's.

For future works, in order not to ignore the difference of the phase between the mixed and separated audio, we intend to model in the waveform domain or use the Griffin-Lim algorithm [22].

6. ACKNOWLEDGEMENT

This research was funded in part by the National Natural Science Foundation of China (61773413), Natural Science Foundation of Guangzhou City (201707010363), Six Talent Peaks project in Jiangsu Province (JY-074), Science and Technology Program of Guangzhou City (201903010040).

[1] https://github.com/f90/Wave-U-Net
[2] https://tinglok.github.io

7. REFERENCES

[1] Emmanuel Vincent, "Musical source separation using time-frequency source priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 91–98, 2005.

[2] Pritish Chandna, Marius Miron, Jordi Janer, and Emilia Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *International conference on latent variable analysis and signal separation*. Springer, 2017, pp. 258–266.

[3] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *International Society for Music Information Retrieval (ISMIR)*, 2014, pp. 477–482.

[4] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde, "Singing voice separation with deep u-net convolutional networks," in *International Society for Music Information Retrieval (ISMIR)*, 2017, pp. 745–751.

[5] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv preprint arXiv:1609.03499*, 2016.

[6] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[7] Emad M Grais, Dominic Ward, and Mark D Plumbley, "Raw multi-channel audio source separation using multi-resolution convolutional auto-encoders," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1577–1581.

[8] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 4898–4906.

[9] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Networks*, IEEE Press, chapter 7, 2001.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 61–65.

[12] Yi Luo, Zhuo Chen, and Nima Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[13] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.

[14] Derry Fitzgerald and Rajesh Jaiswal, "On the use of masking filters in sound source separation," in *the 15th Int. Conference on Digital Audio Effects (DAFx-12)*. 2012.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "MUSDB18 - a corpus for music separation," *10.5281/zenodo.1117371, hal-02190845*, 2017.

[19] Antoine Liutkus, Derry Fitzgerald, and Zafar Rafii, "Scalable audio separation with light kernel additive modelling," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 76–80.

[20] Sanjay Surendranath Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Software available from tensorflow. org*, 2016.

[21] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[22] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.