TASNET: TIME-DOMAIN AUDIO SEPARATION NETWORK FOR REAL-TIME, SINGLE-CHANNEL SPEECH SEPARATION

Yi Luo *Nima Mesgarani*

Department of Electrical Engineering, Columbia University, New York, NY

ABSTRACT

Robust speech processing in multi-talker environments requires effective speech separation. Recent deep learning systems have made significant progress toward solving this problem, yet it remains challenging particularly in real-time, short latency applications. Most methods attempt to construct a mask for each source in time-frequency representation of the mixture signal which is not necessarily an optimal representation for speech separation. In addition, time-frequency decomposition results in inherent problems such as phase/magnitude decoupling and long time window which is required to achieve sufficient frequency resolution. We propose Time-domain Audio Separation Network (TasNet) to overcome these limitations. We directly model the signal in the time-domain using an encoder-decoder framework and perform the source separation on nonnegative encoder outputs. This method removes the frequency decomposition step and reduces the separation problem to estimation of source masks on encoder outputs which is then synthesized by the decoder. Our system outperforms the current state-of-the-art causal and noncausal speech separation algorithms, reduces the computational cost of speech separation, and significantly reduces the minimum required latency of the output. This makes TasNet suitable for applications where low-power, real-time implementation is desirable such as in hearable and telecommunication devices.

Index Terms— Source separation, single channel, raw waveform, deep learning

1. INTRODUCTION

Real-world speech communication often takes place in crowded, multi-talker environments. A speech processing system that is designed to operate in such conditions needs the ability to separate speech of different talkers. This task which is effortless for humans has proven very difficult to model in machines. In recent years, deep learning approaches have significantly advanced the state of this problem compared to traditional methods [1, 2, 3, 4, 5, 6].

A typical neural network speech separation algorithm starts with calculating the short-time Fourier transform (STFT) to create a time-frequency (T-F) representation of the mixture sound. The T-F bins that correspond to each source are then separated, and are used to synthesize the source waveforms using inverse STFT. Several issues arise in this framework. First, it is unclear whether Fourier decomposition is the optimal transformation of the signal for speech separation. Second, because STFT transforms the signal into a complex domain, the separation algorithm needs to deal with both magnitude and the phase of the signal. Because of the difficulty in modifying the phase, the majority of proposed methods only modify the magnitude of the STFT by calculating a time-frequency mask for each source, and synthesize using the masked magnitude spectrogram with the original phase of the mixture. This imposes an upper bound on separation performance. Even though several systems have been proposed to use the phase information to design the masks, such as the phase-sensitive mask [7] and complex ratio mask [8], the upper bound still exists since the reconstruction is not exact. Moreover, effective speech separation in STFT domain requires high frequency resolution which results in relatively large time window length, which is typically more than 32 ms for speech [3, 4, 5] and more than 90 ms for music separation [9]. Because the minimum latency of the system is bounded by the length of the STFT time window, this limits the use of such systems when very short latency is required, such as in telecommunication systems or hearable devices.

A natural way to overcome these obstacles is to directly model the signal in the time-domain. In recent years, this approach has been successfully applied in tasks such as speech recognition, speech synthesis and speech enhancement [10, 11, 12, 13, 14], but waveform-level speech separation with deep learning has not been investigated yet. In this paper, we propose Time-domain Audio Separation Network (TasNet), a neural network that directly models the mixture waveform using an encoder-decoder framework, and performs the separation on the output of the encoder. In this framework, the mixture waveform is represented by a nonnegative weighted sum of N basis signals, where the weights are the outputs of the encoder, and the basis signals are the filters of the decoder. The separation is done by estimating the weights that correspond to each source from the mixture weight. Because the weights are nonnegative, the estimation of source weights can be formulated as finding the masks which indicate the contribution of each source to the mixture weight, similar to the T-F masks that are used in STFT systems. The source waveforms are then reconstructed using the learned decoder.

This signal factorization technique shares the motivation behind independent component analysis (ICA) with nonnegative mixing matrix [15] and semi-nonnegative matrix factorization (semi-NMF) [16]. However unlike ICA or semi-NMF, the weights and the basis signals are learned in a nonnegative autoencoder framework [17, 18, 19, 20], where the encoder is a 1-D convolutional layer and the decoder is a 1-D deconvolutional layer (also known as transposed convolutional). In this scenario, the mixture weights replace the commonly used STFT representations.

Since TasNet operates on waveform segments that can be as small as 5 ms, the system can be implemented in real-time with very low latency. In addition to having lower latency, TasNet outperforms the state-of-art STFT-based system. In applications that do not require real-time processing, a noncausal separation module can also be used to further improve the performance by using information from the entire signal.

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

2. MODEL DESCRIPTION

2.1. Problem formulation

The problem of single-channel speech separation is formulated as estimating C sources $s_1(t), \dots, s_c(t)$, given the discrete waveform of the mixture $x(t)$

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

We first segment the mixture and clean sources into K non-overlapping vectors of length L samples, $\mathbf{x}_k \in \mathbb{R}^{1\times L}$ (note that K varies from utterance to utterance)

$$\begin{cases} \mathbf{x}_k = x(t) \\ \mathbf{s}_{i,k} = s_i(t) \end{cases} \quad t \in [kL, (k+1)L),\ k = 1, 2, \dots, K \tag{2}$$

For simplicity, we drop the notation k where there is no ambiguity. Each segment of mixture and clean signals can be represented by a nonnegative weighted sum of N basis signals $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N] \in \mathbb{R}^{N\times L}$

$$\begin{cases} \mathbf{x} = \mathbf{w}\mathbf{B} \\ \\ \mathbf{s}_i = \mathbf{d}_i\mathbf{B} \end{cases} \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^{1\times N}$ is the mixture weight vector, and $\mathbf{d}_i \in \mathbb{R}^{1\times N}$ is the weight vector for the source i. Separating the sources in this representation is then reformulated as estimating the weight matrix of each source $\mathbf{d}_i \in \mathbb{R}^{1\times N}$ given the mixture weight $\mathbf{w}$, subject to:

$$\mathbf{w} = \sum_{i=1}^{C} \mathbf{d}_i \tag{4}$$

Because all weights $(\mathbf{w}, \mathbf{d}_i)$ are nonnegative, estimating the weight of each source can be thought of as finding its corresponding mask-like vector, $\mathbf{m}_i$, which is applied to the mixture weight, $\mathbf{w}$, to recover $\mathbf{D}_i$:

$$\mathbf{w} = \sum_{i=1}^{C} \mathbf{w} \odot (\mathbf{d}_i \oslash \mathbf{w}) := \mathbf{w} \odot \sum_{i=1}^{C} \mathbf{m}_i \tag{5}$$

$$\mathbf{d}_i = \mathbf{m}_i \odot \mathbf{w} \tag{6}$$

where $\mathbf{m}_i \in \mathbb{R}^{1\times N}$ represents the relative contribution source i to the mixture weight matrix, and $\odot$ and $\oslash$ denotes element-wise multiplication and division.

In comparison to other matrix factorization algorithms such as ICA where the basis signals are required to have distinct statistical properties or explicit frequency band preferences, no such constraints are imposed here. Instead, the basis signals are jointly optimized with the other parameters of the separation network during training. Moreover, the synthesis of the source signal from the weights and basis signals is done directly in the time-domain, unlike the inverse STFT step which is needed in T-F based solutions.

2.2. Network design

Figure 1 shows the structure of the network. It contains three parts: an encoder for estimating the mixture weight, a separation module, and a decoder for source waveform reconstruction. The combination of the encoder and the decoder modules construct a nonnegative autoencoder for the waveform of the mixture, where the nonnegative weights are calculated by the encoder and the basis signals are the 1-D filters in the decoder. The separation is performed on the mixture weight matrix using a subnetwork that estimates a mask for each source.

2.2.1. *Encoder for mixture weight calculation*

The estimation of the nonnegative mixture weight $\mathbf{w}_k$ for segment k is done by a 1-D gated convolutional layer

$$\mathbf{w}_k = ReLU(\mathbf{x}_k \circledast \mathbf{U}) \odot \sigma(\mathbf{x}_k \circledast \mathbf{V}), \quad k = 1, 2, \dots, K \tag{7}$$

where $\mathbf{U} \in \mathbb{R}^{N\times L}$ and $\mathbf{V} \in \mathbb{R}^{N\times L}$ are N vectors with length L, and $\mathbf{w}_k \in \mathbb{R}^{1\times N}$ is the mixture weight vector. σ denotes the Sigmoid activation function and $\circledast$ denotes convolution operator. $\mathbf{x}_k \in \mathbb{R}^{1\times L}$ is the k-th segment of the entire mixture signal $x(t)$ with length L, and is normalized to have unit L^2 norm to reduce the variability. The convolution is applied on the rows (time dimension).

This step is motivated by the gated CNN approach that is used in language modeling [21], and empirically it performs significantly better than using only ReLU or Sigmoid in our system.

2.2.2. *Separation network*

The estimation of the source masks is done with a deep LSTM network to model the time dependencies across the K segments, followed by a fully-connected layer with Softmax activation function for mask generation. The input to the LSTM network is the sequence of K mixture weight vectors $\mathbf{w}_1, \dots \mathbf{w}_K \in \mathbb{R}^{1\times N}$, and the output of the network for source i is K mask vectors $\mathbf{m}_{i,1}, \dots, \mathbf{m}_{i,K} \in \mathbb{R}^{1\times N}$. The procedure for estimation of the masks is the same as the T-F mask estimation in [4], where a set of masks are generated by several LSTM layers followed by a fully-connected layer with Softmax function as activation.

To speed up and stabilize the training process, we normalize the mixture weight vector $\mathbf{w}_k$ in a way similar to layer normalization [22]

$$\hat{\mathbf{w}}_k = \frac{\mathbf{g}}{\sigma} \otimes (\mathbf{w}_k - \mu) + \mathbf{b}, \quad k = 1, 2, \dots, K \tag{8}$$

$$\mu = \frac{1}{N}\sum_{j=1}^{N} \mathbf{w}_{k,j} \quad \sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\mathbf{w}_{k,j} - \mu)^2} \tag{9}$$

where parameters $\mathbf{g} \in \mathbb{R}^{1\times N}$ and $\mathbf{b} \in \mathbb{R}^{1\times N}$ are gain and bias vectors that are jointly optimized with the network. This normalization step results in scale invariant mixture weight vectors and also enables more efficient training of the LSTM layers.

Starting from the second LSTM layer, an identity skip connection [23] is added between every two LSTM layers to enhance the gradient flow and accelerate the training process.

2.2.3. *Decoder for waveform reconstruction*

The separation network produces a mask matrix for each source i $\mathbf{M}_i = [\mathbf{m}_{i,1}, \dots, \mathbf{m}_{i,K}] \in \mathbb{R}^{K\times N}$ from the mixture weight $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_K] \in \mathbb{R}^{K\times N}$ across all the K segments. The source weight matrices can then be calculated by

$$\mathbf{D}_i = \mathbf{W} \odot \mathbf{M}_i \tag{10}$$

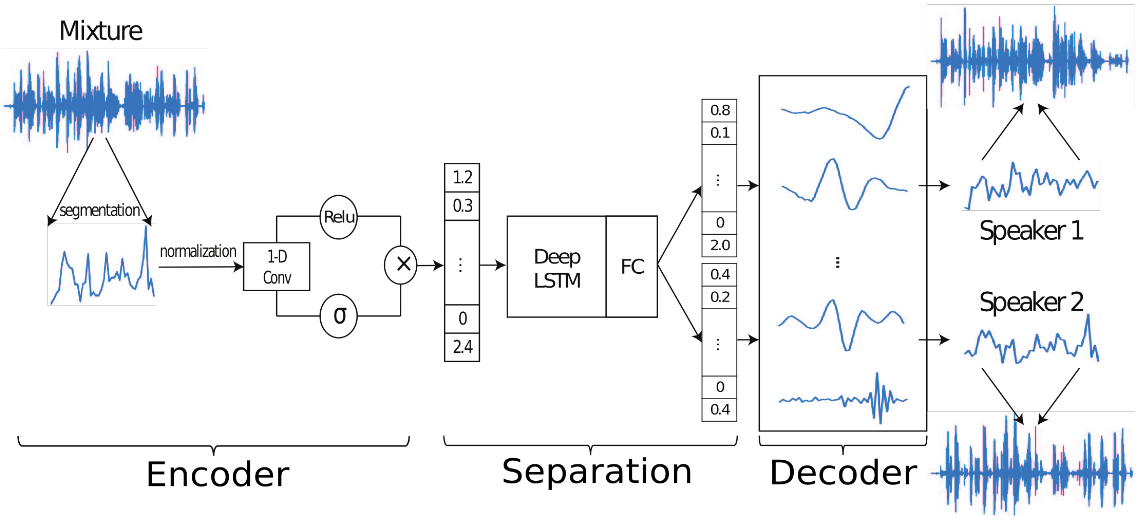


Fig. 1. Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder.

where $\mathbf{D}_i = [\mathbf{d}_{i,1}, \ldots, \mathbf{d}_{i,K}] \in \mathbb{R}^{K\times N}$ is the weight matrix for source i. Note that $\mathbf{M}_i$ is applied to the original mixture weight $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ instead of normalized weight $\hat{\mathbf{W}}$. The time-domain synthesis of the sources is done by matrix multiplication between $\mathbf{D}_i$ and the basis signals $\mathbf{B} \in \mathbb{R}^{N\times L}$

$$\mathbf{S}_i = \mathbf{D}_i\mathbf{B} \tag{11}$$

For each segment, this operation can also be formulated as a linear deconvolutional operation (also known as transposed convolution) [24], where each row in $\mathbf{B}$ corresponds to a 1-D filter which is jointly learned together with the other parts of the network. This is the inverse operation of the convolutional layer in Section 2.2.1.

Finally we scale the recovered signals to reverse the effect of L^2 normalization of $\mathbf{x}_k$ discussed in Section 2.2.1. Concatenating the recoveries across all segments reconstruct the entire signal for each source.

$$s_i(t) = [\mathbf{S}_{i,k}], \quad k = 1, 2, \ldots, K \tag{12}$$

2.2.4. *Training objective*

Since the output of the network are the waveforms of the estimated clean signals, we can directly use source-to-distortion ratio (SDR) as our training target. Here we use scale-invariant source-to-noise ratio (SI-SNR), which is used as the evaluation metric in place of the standard SDR in [3, 5], as the training target. The SI-SNR is defined as:

$$\mathbf{s}_{target} = \frac{\langle \hat{\mathbf{s}}, \mathbf{s}\rangle \mathbf{s}}{\|\mathbf{s}\|^2} \tag{13}$$

$$\mathbf{e}_{noise} = \hat{\mathbf{s}} - \mathbf{s}_{target} \tag{14}$$

$$\text{SI-SNR} := 10\,log_{10}\frac{\|\mathbf{s}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2} \tag{15}$$

where $\hat{\mathbf{s}} \in \mathbb{R}^{1\times t}$ and $\mathbf{s} \in \mathbb{R}^{1\times t}$ are the estimated and target clean sources respectively, t denotes the length of the signals, and $\hat{\mathbf{s}}$ and $\mathbf{s}$ are both normalized to have zero-mean to ensure scale-invariance. Permutation invariant training (PIT) [4] is applied during training to remedy the source permutation problem [3, 4, 5].

3. EXPERIMENTS

3.1. Dataset

We evaluated our system on two-speaker speech separation problem using WSJ0-2mix dataset [3, 4, 5], which contains 30 hours of training and 10 hours of validation data. The mixtures are generated by randomly selecting utterances from different speakers in Wall Street Journal (WSJ0) training set si_tr_s, and mixing them at random signal-to-noise ratios (SNR) between 0 dB and 5 dB. Five hours of evaluation set is generated in the same way, using utterances from 16 unseen speakers from si_dt_05 and si_et_05 in the WSJ0 dataset. To reduce the computational cost, the waveforms were down-sampled to 8 kHz.

3.2. Network configuration

The parameters of the system include the segment length L, the number of basis signals N, and the configuration of the deep LSTM separation network. Using a grid search, we found optimal L to be 40 samples (5 ms at 8 kHz) and N to be 500. We designed a 4 layer deep uni-directional LSTM network with 1000 hidden units in each layer, followed by a fully-connected layer with 1000 hidden units that generates two 500-dimensional mask vectors. For the noncausal configuration with bi-directional LSTM layers, the number of hidden units in each layer is set to 500 for each direction. An identical skip connection is added between the output of the second and last LSTM layers.

During training, the batch size is set to 128, and the initial learning rate is set to $3e^{-4}$ for the causal system (LSTM) and $1e^{-3}$ for the noncausal system (BLSTM). We halve the learning rate if the accuracy on validation set is not improved in 3 consecutive epochs.

Table 1. SI-SNR (dB) and SDR (dB) for different methods on WSJ0-2mix dataset.

Method	Causal	SI-SNRi	SDRi
uPIT-LSTM [4]	✓	–	7.0
TasNet-LSTM	✓	7.7	**8.0**
DPCL++ [3]	×	**10.8**	–
DANet [5]	×	10.5	–
uPIT-BLSTM-ST [4]	×	–	10.0
TasNet-BLSTM	×	**10.8**	**11.1**

The criteria for early stopping is no decrease in the cost function on the validation set for 10 epochs. Adam [25] is used as the optimization algorithm. No further regularization or training procedures were used.

We apply curriculum training strategy [26] in a similar fashion with [3, 5]. We start the training the network on 0.5 second long utterances, and continue training on 4 second long utterances afterward.

3.3. Evaluation metrics

For comparison with previous studies, we evaluated our system with both SI-SNR improvement (SI-SNRi) and SDR improvement (SDRi) metrics used in [3, 4, 5], where the SI-SNR is defined in Section 2.2.4, and the standard SDR is proposed in [27].

3.4. Results and analysis

Table 1 shows the performance of our system as well as three state-of-art deep speech separation systems, Deep Clustering (DPCL++, [3]), Permutation Invariant Training (PIT, [4]), and Deep Attractor Network (DANet, [5]). Here TasNet-LSTM represents the causal configuration with uni-directional LSTM layers. TasNet-BLSTM corresponds to the system with bi-directional LSTM layers which is noncausal and cannot be implemented in real-time. For the other systems, we show the best performance reported on this dataset.

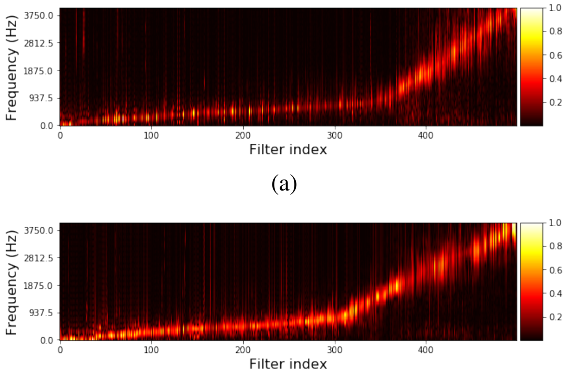
We see that with causal configuration, the proposed TasNet system significantly outperforms the state-of-art causal system which uses a T-F representation as input. Under the noncausal configuration, our system outperforms all the previous systems, including the two-stage systems DPCL++ and uPIT-BLSTM-ST which have a second-stage enhancement network. Note that our system does not contain any regularizers such as recurrent dropout (DPCL++) or post-clustering steps for mask estimation (DANet).

Table 2 compares the latency of different causal systems. The latency of a system T_{tot} is expressed in two parts: T_i is the initial delay of the system that is required in order to receive enough samples to produce the first output. T_p is the processing time for a segment, estimated as the average per-segment processing time across the entire test set. The model was pre-loaded on a Titan X Pascal GPU before the separation of the first segment started. The average processing speed per segment in our system is less than 0.23 ms, resulting in a total system latency of 5.23 ms. In comparison, a STFT-based system requires at least 32 ms time interval to start the processing, in addition to the processing time required for calculation of STFT, separation, and inverse STFT. This enables our system to preform in situation that can tolerate only short latency, such as hearing devices and telecommunication applications.

Table 2. Minimum latency (ms) of causal methods.

Method	T_i	T_p	T_{tot}
uPIT-LSTM [4]	32	–	>32
TasNet-LSTM	5	0.23	**5.23**

To investigate the properties of the basis signals $\mathbf{B}$, we visualized the magnitude of their Fourier transform in both causal and noncausal networks. Figure 2 shows the frequency response of the basis signals sorted by their center frequencies (i.e. the bin index corresponding to the the peak magnitude). We observe a continuous transition from low to high frequency, showing that the system has learned to perform a spectral decomposition of the waveform, similar to the finding in [10]. We also observe that the frequency bandwidth increases with center frequency similar to mel-filterbanks. In contrast, the basis signals in TasNet have a higher resolution in lower frequencies compared to Mel and STFT. In fact, 60% of the basis signals have center frequencies below 1 kHz (Fig. 2), which may indicate the importance of low-frequency resolution for accurate speech separation. Further analysis of the network representation and transformation may lead to better understanding of how the network separates competing speakers [28].



(a)

(b)

Fig. 2. Frequency response of basis signals in (a) causal and (b) noncausal networks.

4. CONCLUSION

In this paper, we proposed a deep learning speech separation system that directly operates on the sound waveforms. Using an autoencoder framework, we represent the waveform as nonnegative weighted sum of a set of learned basis signals. The time-domain separation problem then is solved by estimating the source masks that are applied to the mixture weights. Experiments showed that our system was 6 times faster compared to the state-of-art STFT-based systems, and achieved significantly better speech separation performance. Audio samples are available at [29].

5. ACKNOWLEDGEMENT

This work was funded by a grant from National Institute of Health, NIDCD, DC014279, National Science Foundation CAREER Award, and the Pew Charitable Trusts.

6. REFERENCES

[1] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 12, pp. 2136–2147, 2015.

[2] Xiao-Lei Zhang and DeLiang Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 5, pp. 967–977, 2016.

[3] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *Interspeech 2016*, pp. 545–549, 2016.

[4] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

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

[6] 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.

[7] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 708–712.

[8] Donald S Williamson, Yuxuan Wang, and DeLiang Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 24, no. 3, pp. 483–492, 2016.

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

[10] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] Pegah Ghahremani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "Acoustic modelling from the signal domain using cnns.," in *INTERSPEECH*, 2016, pp. 3434–3438.

[12] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[13] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[14] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "Segan: Speech enhancement generative adversarial network," *Proc. Interspeech 2017*, pp. 3642–3646, 2017.

[15] Fa-Yu Wang, Chong-Yung Chi, Tsung-Han Chan, and Yue Wang, "Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 875–888, 2010.

[16] Chris HQ Ding, Tao Li, and Michael I Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

[17] Ehsan Hosseini-Asl, Jacek M Zurada, and Olfa Nasraoui, "Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2486–2498, 2016.

[18] Andre Lemme, René Felix Reinhart, and Jochen Jakob Steil, "Online learning and generalization of parts-based image representations by non-negative sparse autoencoders," *Neural Networks*, vol. 33, pp. 194–203, 2012.

[19] Jan Chorowski and Jacek M Zurada, "Learning understandable neural networks with nonnegative weight constraints," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 1, pp. 62–69, 2015.

[20] Paris Smaragdis and Shrikant Venkataramani, "A neural network alternative to non-negative audio models," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 86–90.

[21] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *International Conference on Machine Learning*, 2017, pp. 933–941.

[22] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[24] Vincent Dumoulin and Francesco Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[25] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[27] 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.

[28] Tasha Nagamine and Nima Mesgarani, "Understanding the representation and computation of multilayer perceptrons: A case study in speech recognition," in *International Conference on Machine Learning*, 2017, pp. 2564–2573.

[29] "Audio samples for TasNet," `http://naplab.ee.columbia.edu/tasnet.html`.