Efficacy of Orthogonal Embedding Matrix in Deep Clustering for Single-Channel Speech Separation

Soyeon Choe

Department of Electrical and Electronic Engineering

The Graduate School

YONSEI University

Efficacy of Orthogonal Embedding Matrix in Deep Clustering for Single-Channel Speech Separation

Soyeon Choe

A Dissertation

Submitted to the Department of Electrical and Electronic Engineering

and the Graduate School of Yonsei University

in partial fulfillment of the

requirements for the degree of

Master of Science

Department of Electrical and Electronic Engineering
The Graduate School

YONSEI University
Seoul, KOREA

December 2018

This certifies that the dissertation
of Soyeon Choe is approved.

Supervisor: Hong-Goo Kang

Prof. Chungyong Lee

Dr. Youna Ji

The Graduate School
Yonsei University
December 2018

ACKNOWLEDGEMENT

I would like to express my deep gratitude to Prof. Hong-Goo Kang for his valuable and constructive suggestions during the planning and development of this research work. I would like to thank Prof. Chungyong Lee and Dr. Youna Ji for being part of my defense and giving me invaluable comments and suggestions to improve my dissertation. I would also like to thank my colleagues in DSP-AI Lab. for their support and advices throughout my time in the laboratory.

Special thanks to my family, who have always been there for me in good or bad times, and I would like to thank everyone, who has supported me through prayer and encouragement. Finally, I thank God for the guidance of my life and all the blessings He has given me. I hope to always remember His love and live with gratitude. Thank You.

Contents

List of Figures

List of Tables

ABSTRACT

Efficacy of Orthogonal Embedding Matrix in Deep Clustering for Single-Channel Speech Separation

Soyeon Choe
Dept. of Electrical and Electronic Engineering
The Graduate School, Yonsei University, Seoul, Korea

This dissertation improves the quality of deep learning-based speech separation algorithm by extending the conventional deep clustering. Deep clustering is a deep neural network-based speech separation algorithm that first trains the mixed component of signals with high-dimensional embeddings, and then uses a clustering algorithm to separate each mixture of sources. As an extension of the deep clustering method, deep attractor network is also introduced and analyzed.

In the training procedure, bi-directional long short-term memory (BLSTM) is used to learn the embeddings that contain information of mixed signals. In this dissertation, the baseline criterion of deep clustering is extended with an additional regularization

term to further improve the overall performance. This term plays a role in assigning a condition to the embeddings such that it gives less correlation to each embedding dimension, leading to better decomposition of the spectral bins. The regularization term helps to mitigate the unavoidable permutation problem in the conventional deep clustering method, which enables to bring better clustering through the formation of optimal embeddings.

With the application of K-means clustering in inference procedure, the optimization of embedding decomposition is an essential factor to obtain more desirable clustered output. With the assumption that each embedding matrix works as a basis, less correlation of embeddings results to achieve more independence among themselves. Therefore, the permutation problem of deep clustering method is ameliorated with the use of penalization term by better clustering of the embeddings. Also, the result is evaluated by investigating the performance with varying embedding dimension, signal to interference ratio (SIR), and gender dependency. Comparison in performance with the source separation measurement metrics, *i.e.* signal to distortion ratio (SDR), improved normalized projection alignment (NPA), and relative error rate, confirm that the proposed method outperforms the conventional deep clustering method.

Key words : Speech enhancement, deep clustering, deep attractor network, penalization term, embedding matrix, single-channel speech separation, deep learning, deep neural network (DNN), bi-directional long short-term memory (Bi-LSTM)

Chapter 1

Introduction

Since the world is full of various sounds such as environmental sound, human sound, and more, most humans hear more than one audio sources at once. In this case, human can automatically distinguish various sources, *e.g.* recognizing the voice of a friend in a very noisy environment, distinguishing various instruments in a song, talking on the phone with a noisy background, etc. Mesgarani and Chang [1] say this separating process is unconsciously performed through the brain without anyone noticing. As this process is naturally done in human body, people wondered if this model could be made mathematically. With the success of speech separation, it could help solve some critical limitations in various audio processing techniques, especially in automatic speech recognition (ASR). Even though the ASR system reaches the human level in matched conditions [2], the performance is not as great under more complex conditions [3].

In recent years, the accuracy of ASR system in real-world applications has been essentially improved; however, there are still some difficulties in solving the cocktail party problem [4][5], *i.e.* recognizing the speech of the specific speaker among multiple

speakers along with various background noises. Although this seemed easy to solve for humans, it was a difficult problem to be solved mathematically. In order to solve this kind of problem, a lot of people researched to improve the front-end of ASR system through pre-prosessing and applied the enhanced signals as the input of the recognition system. The basic scheme of the front-end of speech recognition is shown in Figure 1.1. Highly used pre-processing method is speech separation, which separates mixed sources into target and interfering sources.

One of the conventional speech separation techniques proposed decades ago is computational auditory scene analysis (CASA) [6][7], which studies how humans separate speech and learn from them. However, some big drawbacks are the manually designed rules due to limited number of observations and the limitations from not learning the data automatically [8]. Another speech separation technique is non-negative matrix factorization (NMF) [9][10], which uses hand-designed rules from human observations with the assumption that audio spectrogram has a low rank structure that can be represented with a small number of bases [11]. Even though this technique seems successful, the speaker dependency issue of the basis leads to some constraints, and there are also some limitations in real-time applications due to the complexity of the decomposition and the number of basis. Also, another approach of speech separation is adopting multi-channel techniques using a beamforming technique [12]. Beamformer is a spatial filter that operates on the outputs of a microphone array and forms a beam pattern to enhance the desired speech coming from one direction while suppressing interfering speech or noise from other directions [13]. However since the technique requires multiple microphones, it is more complex than single-channel source separation methods.

With the increase of deep learning applications in various fields, how to make good use of deep learning technique in speech separation became an essential factor in speech recognition. One example is separation with supervised regression, which directly estimates magnitude spectra of each source or estimates a set of masks and reconstructs the magnitude. Especially in single-channel, separating and recognizing speech in cocktail party condition was highly challenging problem for decades, until the recent release of single-channel separation method, *deep clustering* [14][15]. Hershey et al.[14] proposed a speech separation framework using embedding matrix as a key factor to cluster mixed signals by training a neural network to assign embedding vector to each element of high-dimensional signal. After short time Fourier transform (STFT) of mixed signals, embeddings are assigned to each time-frequency (T-F) index. Then each T-F bin is assigned to one of the sources through the clustering of embeddings, and these assignments are used to form estimated masks for the mixed signal. Even though deep clustering [14] is somehow helpful to overcome the permutation problem, each separated output still contains undesired segments if there is an error during the clustering process.

In this dissertation, the limitations in single-channel speech separation is discussed in details, and conventional algorithms are reviewed. Moreover, some implementations and extensions from the conventional speech separation models, *i.e.* deep clustering (DC) [14] and deep attractor network (DANet) [16], are introduced with the provision of improved results. This dissertation aims to develop an effective deep clustering framework by designing the training strategy to assign an additional regularization condition. By training the network to find the minimum loss between the embeddings and the diagonal matrix, the network automatically assigns an orthogonal constraint to the em-

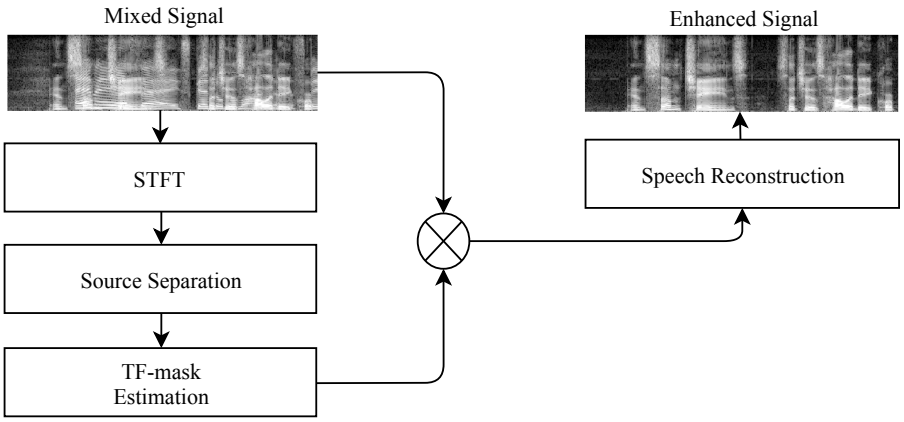


Figure 1.1: Front-end system of speech recognition

beddings. Since each embedding can be considered as a basis of each input speaker, the proposed training method leads to have more embedding independency so that embeddings are useful for better decomposition. This effective formation of embeddings helps to improve the remaining permutation problem of the conventional method as the decomposed spectral bins help increase the performance of clustering. Also, the improvements of the proposed algorithm over the conventional deep clustering algorithm are demonstrated, and the results in various experimental setups are analyzed.

This dissertation consists of five sections as follows. Section 2 describes the details on the deep clustering method followed by the proposed algorithm in section 3. Moreover, the experimental results and conclusion are described in section 4 and section 5, respectively.

Chapter 2

Background

This chapter introduces the background methods and the basic framework of single-channel speech separation. The basic elements in speech separation is dealt in this chapter, along with several conventional source separation methods, such as signal processing-based, rule-based, and decomposition-based methods.

2.1 Audio signal processing

Normally a sound signal is recorded through the microphone, and the signal is sampled into a discrete signal for signal processing. Since more than one sources of sound are recorded at certain time, the mixture of sound can be represented as the equation below:

$$y(t) = \sum_i x_i(t), \tag{2.1}$$

where $y(t)$ is the mixture at time t, and x_i represents each source in index i. After sampling of signals, the discrete signal is represented in time-domain as the equation below:

$$y[n] = \sum_i x_i[n]. \tag{2.2}$$

Moreover, Fourier analysis is most commonly applied for audio signal processing to project the time-domain signal into a space by the sum of sinusoids. The transformed signal is represented with the equation shown below:

$$Y(\omega) = \sum_i X_i(\omega), \tag{2.3}$$

where X_i and Y are each Fourier transformed source and mixture. Since audio signal is continuous for certain period of time, the signal changes over time. Therefore, the signal is assumed to be stationary when processing with short periods of signal, and this process is called short time Fourier transform (STFT). STFT output of the audio mixture is shown in the equation (2.4), where X and Y are Fourier transformed signals, and this transformed output is known as spectrogram. Spectrogram contains time t and frequency index f representing the frame and bin, and each value represents the time-frequency (T-F) bin. Some studies [17] showed that reasonable outcomes could be obtained in speech processing without the phase information, which makes the process become more complex. Therefore, magnitude or power spectrograms are usually used when working with speech, and the equations are respectively shown in equation (2.5) and (2.6) below:

$$Y(f,t) = \sum_i X_i(f,t), \tag{2.4}$$

$$|Y(f,t)| \approx \sum_i |X_i(f,t)|\,, \tag{2.5}$$

$$|Y(f,t)|^2 \approx \sum_i |X_i(f,t)|^2\,. \tag{2.6}$$

2.2 Mask

Mask is one of the most commonly used applications on the speech mixture to obtain the target speech through speech separation. Mask is a matrix with dimension F by T that can mask out the interfering signal by applying it to the mixture spectrogram. Among various types of masks, commonly used masks are binary mask and ratio mask. With the binary mask, it is easy to generate since each T-F bin contains binary value; however, it can not obtain the best separation performance as the source is separated in hard decision. On the other hand, each T-F bin in ratio mask contains values from 0 to 1, which performs in better separation. Examples of masks are shown in Figure 2.1, along with the outputs of their application on the speech mixture. Figure 2.1 indicates that mask with hard decision, *i.e.* ideal binary mask (IBM), contains more distortion in estimated output than the mask with soft decision, *i.e.* ideal ratio mask (IRM).

2.3 Signal processing-based source separation

Signal processing-based source separation is the first algorithm proposed to address the topic of source separation. As shown in [18], speech is usually assumed to follow distributions such as Gaussian or Laplacian and is mainly designed for speech enhancement. In practice, voice activity detector is first applied to the noisy speech, followed by the collection of silent frames to calculate the noise statistics. Then the maximum likelihood optimization is computed to obtain the speech. Since this source separation method can not learn the actual data, it outputs mostly unsatisfying performances. Therefore, developed source separation method followed, as indicated in the next section.

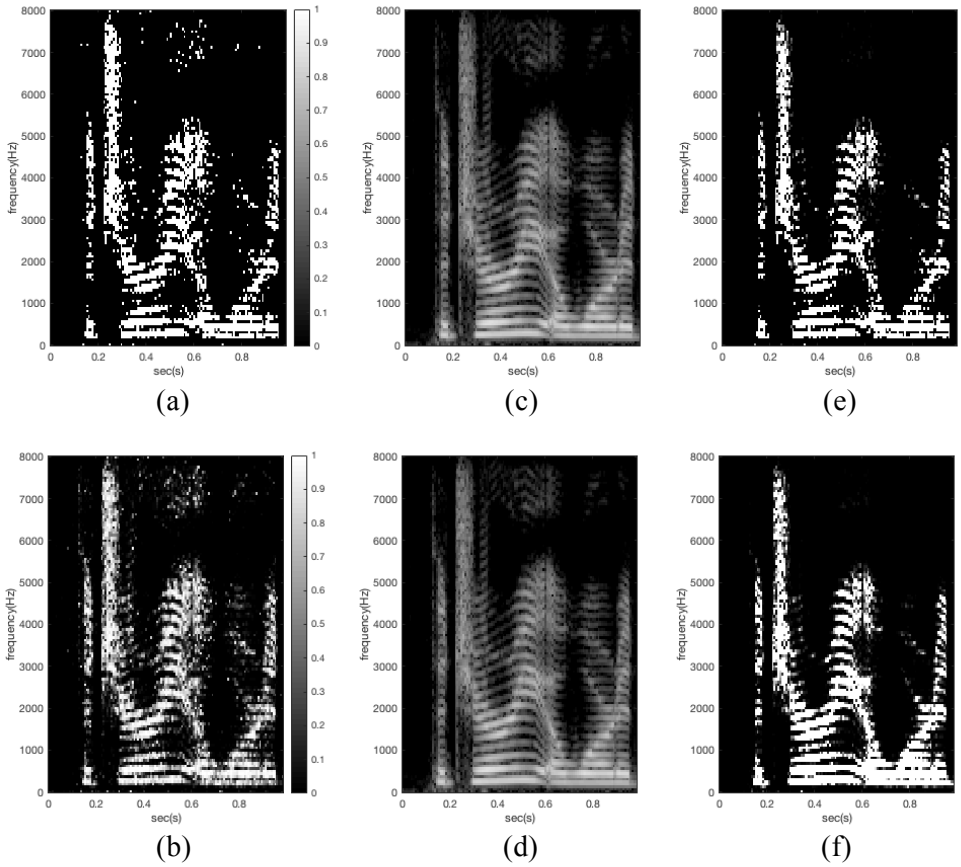


Figure 2.1: Examples of ideal binary mask (IBM) and ideal ratio mask (IRM). (a),(b):IBM and IRM, (c),(d): noisy spectrograms, and (e),(f): estimated outputs with the application of IBM and IRM.

2.4 Rule-based source separation

Based on the properties of speech, researchers [19][20][21][22] proposed rule-based source separation throughout the years, known as the computational auditory scene analysis (CASA). Some of the rules they follow are harmonic, pitch continuity, vocal tract continuity, etc. Assuming that human can only produce one pitch at a time, regions with two or more pitches indicate more than one sources. Therefore among all rules, pitch is usually used them most. In a typical CASA system, a feature by choice is calculated first. Then, the T-F bins are grouped into sources based on hand-designed rules, and lastly, a binary mask is formed based on the T-F bins assignment. Since CASA is limited to work on speech with only one pitch at a time, this became a great drawback. Also, having all rules be hand-designed was another essential limitation to this system.

2.5 Decomposition-based source separation

Unlike the rule-based source separation, which formed the rules with the observation of spectrogram, decomposition-based source separation builds a system that can automatically discover the rules from the data. The assumption of this system is that audio spectrogram has low rank structure, represented with a small number of basis as shown in the equation (2.7):

$$Y = WH, \tag{2.7}$$

where the spectrogram $Y \in \mathbb{R}^{F \times T}$ is decomposed into the matrix product of two matrices $W \in \mathbb{R}^{F \times K}$ and $H \in \mathbb{R}^{K \times T}$. In speech processing, non-negative matrix factorization (NMF) [11] is the most popular decomposition method where W and H are

constrained to be non-negative.

Chapter 3

Neural Network

As slightly mentioned in chapter 1, using deep learning technique in speech separation has become an essential factor to obtain high performance in speech recognition. Therefore, this chapter introduces some fundamental features of the neural network. Basically, artificial neural network (ANN) is a way to simulate the process of human brain with a computer program. ANNs train themselves to learn the knowledge through the detection of various connections and patterns of the data, and the concept of this network is inspired from the biology observation [23].

3.1 Feed-forward network

To begin with the most common neuron named perceptron [24], it is represented with the equation below:

$$o = f(\sum_{k} i_k w_k + b). \tag{3.1}$$

In this equation, o, i_k, w_k, b, and $f(\cdot)$ indicate the output, input, weight, bias, and non-linear function, respectively. With the non-linear function, the weighted sum is converted

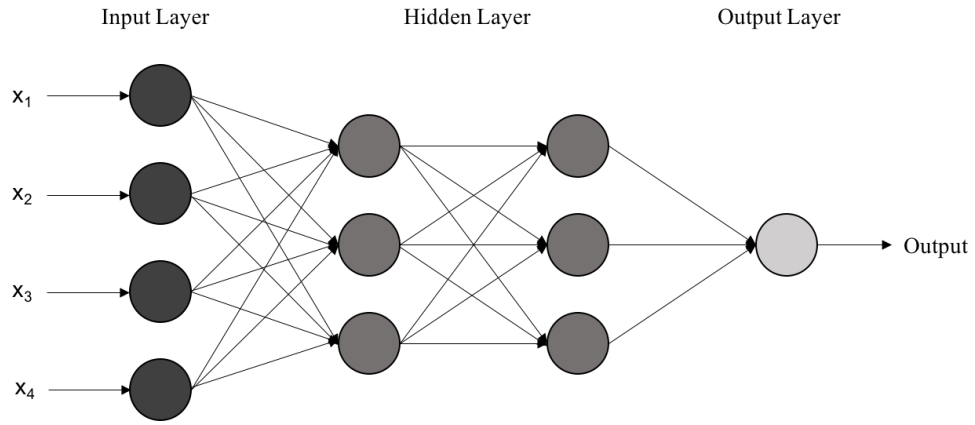


Figure 3.1: Feed forward network with one hidden layer

into various decisions due to the architectures of the network.

The feed-forward network is the first type of ANN that transfers the information in one direction. As indicated in its name, this network moves the information forward, *i.e.* from the input nodes, then through the hidden nodes, and finally toward the output nodes. The simple type of feed-forward network is the perceptron, and it becomes more complex with the addition of hidden units. The concatenation of perceptrons is followed by non-linear functions such as sigmoid, softmax, rectified linear function (ReLU), and more. The system architecture of this network is shown in Figure 3.1.

3.2 Recurrent network

Unlike the feed-forward network that computes the function on fixed size input, recurrent neural network (RNN) learns sequential data with the computation of variable length input, and this system architecture is shown in Figure 3.2. RNN is one of ANNs that

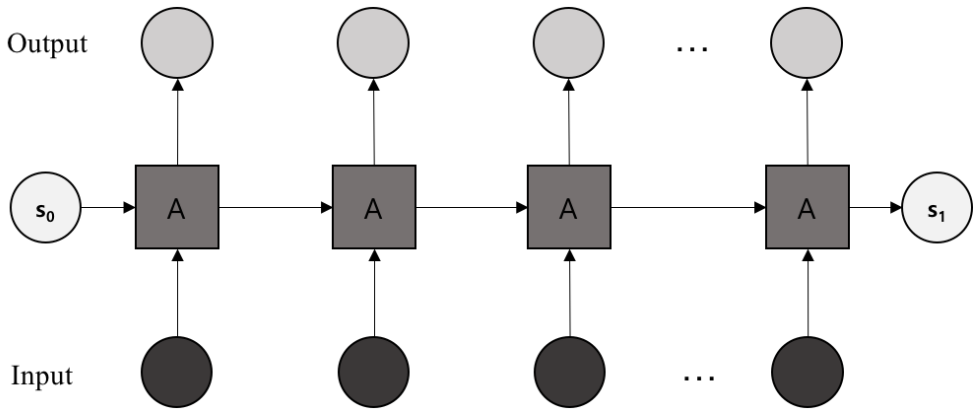


Figure 3.2: System architecture of recurrent neural network

forms a directed cycle with connections between units [25]. This network processes arbitrary input sequences by using their internal memory, and relation between the input and the output is given in equations below in the time step of t:

$$h_t = f(W_h x_t + U h_{t-1} b_h), \tag{3.2}$$

$$y_t = g(W_y h_t + b_y). \tag{3.3}$$

In equation (3.2) and (3.3), h, y, and x are hidden state, output, and input, and W_h, W_y, b_h, and b_y represent the weights and bias of hidden state and output. Also, U indicates the weight between consecutive hidden states with the non-linearity functions denoted as $f(\cdot)$ and $g(\cdot)$. With the additional hidden state passed through time steps, the previous input affects the current output, meaning that this network contains the memory of the past input. Therefore, this network works well in audio processing, since audio is dependent through period of time.

3.3 Long short-term memory

One of the disadvantages of RNN is that it cannot derive the sequences from time steps much far behind, also known as the vanishing gradient problem. In order to fix this problem, a new network is introduced named long short term memory (LSTM) network [26]. This network solves the problem with the cell state and cell unit, which help to remember sequences from far distance. In order to update the cell state, three gates are needed: input gate, output gate, and forget gate. The input and output gate operations are shown in equation (3.4) and (3.5):

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \end{aligned} \tag{3.4}$$

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t * tanh(C_t), \end{aligned} \tag{3.5}$$

which control the flow of the activation by the importance of the new data, and $\tilde{C}_t$ indicates the memory cell. Also, the forget gate function is shown in equation (3.6):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3.6}$$

which controls the memory cell to forget the previous activation with the value between 0 and 1. In summary, f_t, i_t, $\tilde{C}_t$, ω_t, and h_t represent forget gate, input gate, cell state, sigmoid function, and output activation function, respectively. However, this network is limited to learning the values from previous time steps. Therefore as the network also learns from the future time steps, it could achieve better performance, and this is accomplished with the bidirectional RNN (bi-RNN) as shown in Figure 3.3. With bi-

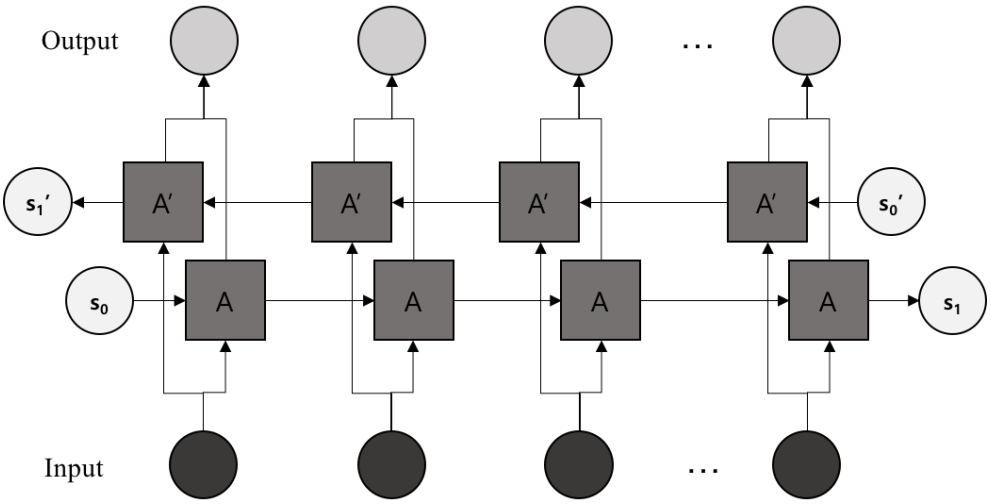


Figure 3.3: System architecture of bidirectional RNN

RNN, the information is propagated in both time directions, and combining bi-RNN and LSTM modules can improve the performance significantly.

3.4 Objective function

For neural network, an objective function is computed to measure the loss of the network for the optimization of the model. Mostly used objective functions are mean squared error (MSE) and cross entropy error, shown in equation (3.7) and (3.8):

$$L = \frac{1}{2}\sum_{k}(y_k - t_k)^2, \tag{3.7}$$

$$L = -\sum_{k} t_t log(y_k). \tag{3.8}$$

For equation (3.7), y_k, t_k, and k refer to the output of the network, the target label, and the dimension of the data, respectively. As the loss between the output and the target decreases, the output is getting closer to the answer.

3.5 Back-propagation

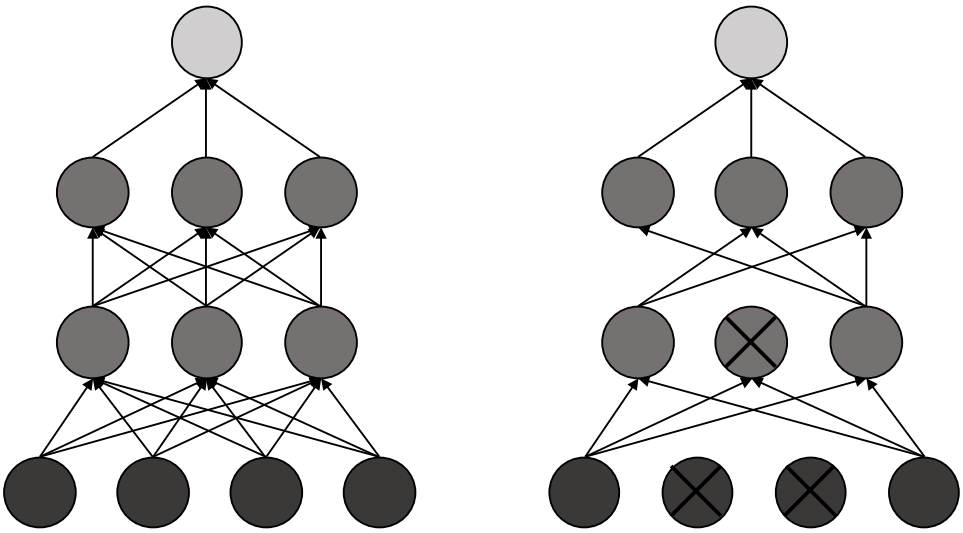
Back-propagation, introduced by [27], is a key algorithm to train ANNs. This algorithm is arranged in two steps: (1) forward propagation of the training data as the input and measurement of the error between the estimated and target values; (2) back propagation of the error to each node in the neural network. For the second step, chain rule is used to update the parameters with gradient-based optimization methods as shown in equation (3.9):

$$\begin{aligned} L &= f(x), \\ \frac{\partial L}{\partial x} &= \frac{\partial L}{\partial f}\frac{\partial f}{\partial x}. \end{aligned} \tag{3.9}$$

3.6 Regularization

In order to control overfitting, regularization is needed in neural networks. Overfitting is defined as the status of improper response to the test data due to over adaptation of the training data, *e.g.* case with a lot of parameters and small amount of training data. A common regularization used in neural network is L^2 regularization which imposes penalty to weight parameters with $\lambda \sum_i \left|\theta_i^2\right|$, where λ is hyperparameter and θ_i is model parameter [28].

Another regularization method is dropout, which randomly drops certain neurons in each layer with the probability of p and trains the remaining neurons [29]. After training, $1/p$ of the weights are remained in each layer, and this prevents the units from having high correlation between the units of a given layer. The standard neural network and the appliance of dropout is shown in Figure 3.4.



(a) Standard neural network (b) Application of dropout

Figure 3.4: Application of dropout on neural network

Chapter 4

Deep Network Speech Separation

In this chapter, several deep learning-based speech separation methods, *i.e.* deep clustering and deep attractor network, are introduced with the proposed algorithm afterwards. The expected efficacy of the proposed method is discussed, followed by the visualization of the network matrix output.

4.1 Deep clustering

Deep clustering is a deep network that uses learned feature transformations known as embeddings, to separate speech on open set of speakers [14]. Deep clustering network presented in [14][15] is reviewed in this dissertation. The main contributions of deep clustering are: (1) use of embeddings to find the appropriate labeling for the input; (2) use of label indicator. In audio signals, raw input signal x can be defined as a feature vector $X_i = g_i(x)(i \in 1, 2, \cdots, N)$, where i is time-frequency index. Deep clustering starts with the assumption that a reasonable partition of elements exists for each region. Its objective is to find that partition using embedding matrix and K-means clustering, in

order to estimate masks to be applied to each input mixture of X. With audio signals, these regions are sets of time-frequency bins, where each source dominates the other source. In order to estimate the partition, the concept of embedding matrix takes place. Deep clustering seeks K-dimensional embeddings of the mixed signal to apply simple clustering in embedding space.

4.1.1 Learning embeddings

In training stage, deep neural network is used to transform input x to K-dimensional embedding $V \in \mathbb{R}^{\{N \times K\}}$, with the embedding considered as unit-norm, $|v_i|^2 = 1$, where v_i is the embedding for element i. Embedding V actually represent *N* by *N* estimated affinity matrix, which analyzes the content of each mixed signal in terms of a set of channels. The affinity matrix VV^T is learned to match YY^T by minimizing the cost using the squared Frobenius norm as follows:

$$C_Y(V) = \left\| VV^T - YY^T \right\|_F^2 . \tag{4.1}$$

YY^T is the target affinity matrix which is generated with the label indicator Y, *i.e.* the concept of comparing each T-F bin and assigning value '1' to the dominant bin, formed as an ideal binary mask (IBM). Note that low-rank formulation leads to efficient implementation of the cost function as shown in the equation below:

$$C_Y(V) = \left\| V^T V \right\|_F^2 - 2 \left\| V^T Y \right\|_F^2 + \left\| Y^T Y \right\|_F^2 . \tag{4.2}$$

4.1.2 Clustering of embeddings

In inference stage, K-means clustering is used to cluster the K-dimensional embeddings into the number of sources. By assigning the value '1' to the spectral bin of each clus-

ter and '0' to the other, ideal binary masks are formed for each source, and applying these masks to the mixed signal separates the signal into individual sources. Therefore, training to obtain appropriate embedding matrix is essential to cluster the embeddings sufficiently, and the overall process of deep clustering is shown in Figure 4.1.

4.1.3 Optimization of training

For deep clustering, LSTM structure is particularly used for RNN with dropout regularization method, which trains regularization by randomly setting some nodes to zero [15]. According to the possibility that dropout interferes with the memorization ability of LSTM, recurrent dropout is used to sample the set of dropout nodes in each sequence for once [30]. In deep clustering, the dropout masks are sampled once at each time step for forward connections and only once for each sequence for recurrent connections [15].

4.1.4 Limitation in deep clustering

Even though deep clustering method has obtained high performance for speech separation, there are also some drawbacks. One of the drawbacks is that the whole training and inference procedure are complex. Also, since clustering is usually occurred only after obtaining the embeddings, it is not suitable for real-time process, and it is difficult to implement end-to-end network. Moreover, even though deep clustering is somehow helpful to overcome the permutation problem, each separated output still contains undesired segments if there is an error during the clustering process.

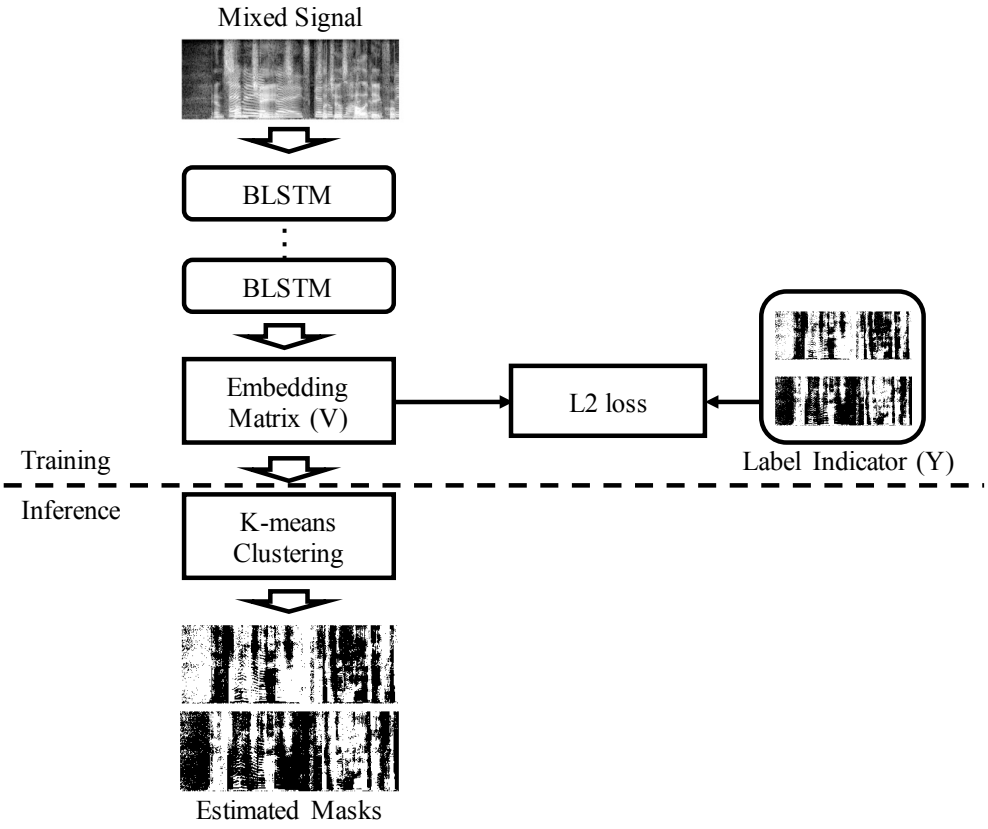


Figure 4.1: System architecture of deep clustering (DC)

4.2 Deep attractor network

Since the main drawback of deep clustering is the mismatch between the training and inference procedure, it is difficult to perform end-to-end network. Therefore, this chapter introduces the subsequent method of speech separation called deep attractor network (DANet) [16]. This network forms an attractor for each source in the embedding space and estimates mask using the similarity between embedded points and each attractor. Since the mask is directly related to the attractor point, this framework can potentially be extended to arbitrary number of sources and the mask learning enables efficient end-to-end training scheme.

4.2.1 Model

DANet is to train the mixture signal X to a K-dimensional embedding space with the objective function below:

$$L = \sum_{f,t,c} \|S_{f,t,c} - X_{f,t} \times M_{f,t,c}\|_2^2 \,, \tag{4.3}$$

where S, X, and M are the clean spectrogram of C sources, mixture spectrogram, and mask, respectively. In order to compute the mask in K-dimensional embedding space, equation (4.4) is used with attractors, $A \in \mathbb{R}^{C\times K}$, and embeddings, $V \in \mathbb{R}^{F\times T\times K}$, where the attractor is defined in equation (4.5) with $Y \in \mathbb{R}^{F\times T\times K}$ as the label indicator:

$$M_{f,t,c} = Sigmoid(\sum_k A_{c,k} \times V_{f,t,k}), \tag{4.4}$$

$$A_{c,k} = \frac{\sum_{f,t} V_{k,f,t} \times Y_{c,f,t}}{\sum_{f,t} Y_{c,f,t}}. \tag{4.5}$$

In order to find the objective function in equation (4.3), embedding V is computed first with a forward pass of neural network. Next, the attractor is estimated for each source with equation (4.5). Then, the mask is estimated with the similarity for each T-F bin in the embedding space with equation (4.4). Since it is likely for source separation to have the summation of each mask equal to one for each T-F bin, the sigmoid function in mask function could be replaced to softmax function as shown in equation (4.6):

$$M_{f,t,c} = Softmax(\sum_{k} A_{c,k} \times V_{f,t,k}). \tag{4.6}$$

Therefore, after computing all values of S, X, and M, L2 loss is used to generate the gradient, and this loss measures the difference between the masked signal and the clean reference. The overall structure of the DANet is shown in Figure 4.2.

4.2.2 Estimation of attractor points

Attractor points can be estimated using various methods other than equation (4.5). For the inference procedure, there are two ways to form the attractor points since the true assignment Y is unknown. First strategy is similar to deep clustering, where the centers are found using K-means clustering, and the second strategy is based on the observation that the location of the attractors in the embedding space is relatively stable [16]. However, the second strategy needs to be tested in more depth.

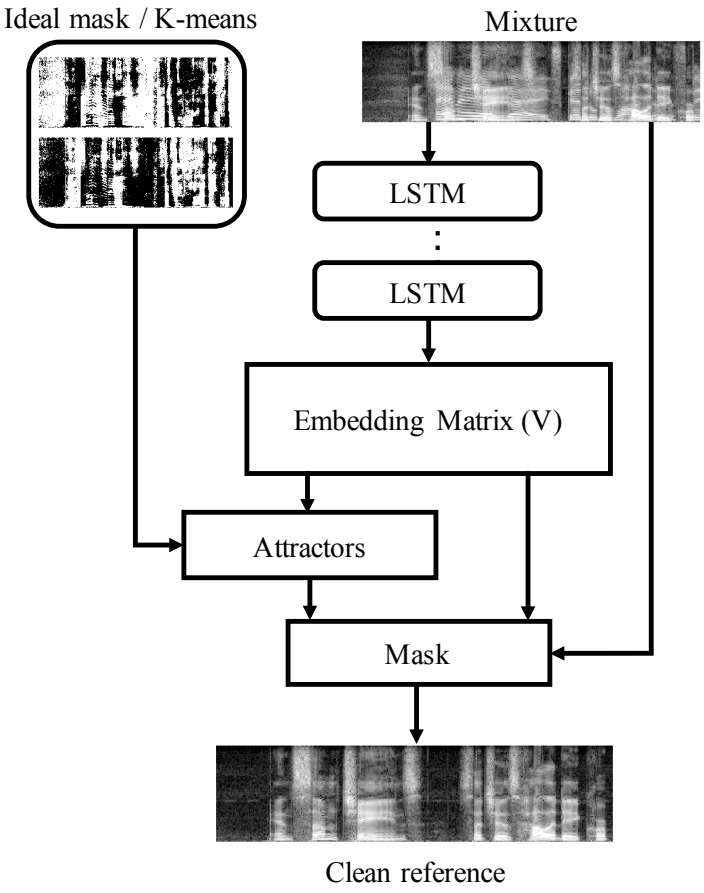


Figure 4.2: System architecture of deep attractor network (DANet)

4.3 Orthogonality of embedding matrix

Even though the deep clustering method obtains high performance in single-channel speech separation tasks, there are still more improvements to be made. Knowing that the formation of optimal embeddings leads to better decomposition of spectral bins, focusing on regularizing the embeddings is more effective to the performance.

4.3.1 Visualization of characteristics of speakers

The characteristics of different speakers were observed to analyze the efficacy of proposed method by plotting the speaker information in 2-dimensional plot. Since information of different gender is decomposed in high-dimension, T-distributed Stochastic Neighbor Embedding (t-SNE) was applied to visualize the speaker matrices in 2-dimension [31]. Points that are close in high-dimensional space remain close in low-dimensional space, and the distance between the points indicates the similarity of each class, *i.e.* closer points means similar characteristics. Observing the t-SNE outputs, the characteristics of the speakers are clearly demonstrated in Figure 4.3. The two figures of Figure 4.3 show the datapoints of different speakers of same and different gender mixtures. Different colors indicate different classes of speakers, *i.e.* speaker 1 and 2 representing target and interfering speakers, respectively. With same gender speaker mixtures in Figure 4.3(a), two different speaker datapoints greatly overlapped each other, verifying that speakers with same gender show similar characteristics. On the other hand, Figure 4.3(b) showed that speakers of different genders have different characteristics with less overlapping datapoints of two speakers. According to the characteristics of speech signals, mixtures with different gender are easier to decompose than those with same

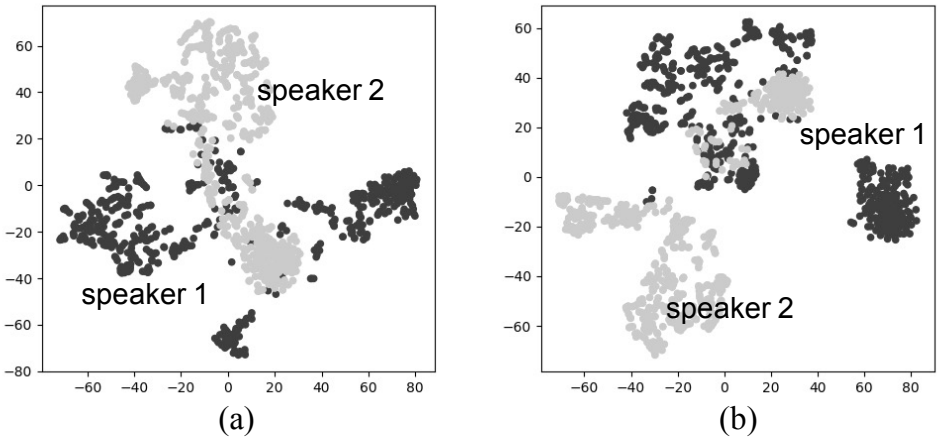


Figure 4.3: t-SNE visualization of mixture signals. (a) Same gender mixture with female target and female interference, (b) different gender mixture with male target and female interference. All mixtures have signal-to-interference ratio (SIR) of 15 dB.

gender. Since female speech signals are in higher frequencies than male speech signals, different gender mixtures have less correlation than same gender mixtures. Therefore, it may be difficult to cluster the mixed signals with different genders than the mixed signals with same genders. Based on these characteristics, the separation performances are analyzed in the next section.

4.3.2 Transition in penalization term

Penalization term is one of the regularization terms to encourage the diversity in annotation vectors as indicated below:

$$P_{orthon} = \left\| V^T V - I \right\|_F^2, \tag{4.7}$$

where *V, I*, and $\|\cdot\|_F$ are embeddings, identity matrix, and Frobenius norm, respectively. Computing V^TV sets the output dimension to become embedding dimension by embedding dimension, *i.e.* $\mathbb{R}^{K\times K}$, so that it is easier to compute the loss between the identity matrix, *i.e.* $I \in \mathbb{R}^{K\times K}$, and V^TV. This term forces the diagonal elements of the embedding matrix to 1 and the off-diagonal to 0, making the embedding matrix to be orthonormal. Applying this term on deep clustering embeddings will lead to assign more independence to each other, resulting in efficient clustering. Like the implementation of the cost function of deep clustering in Equation (4.2), Equation (4.7) can be shown as the equation below:

$$P_{orthon} = \left\|VV^T\right\|_F^2 - 2\left\|V\right\|_F^2 + \left\|I\right\|_F. \tag{4.8}$$

Even though this regularization term improves the speech separation performance according to the result in chapter 5, it is difficult to train the values in the embeddings to become only 0 or 1. Since the diagonal term is much greater than 1, the network has to force the terms to become close to 1, and this step eventually leads to the distortion in the clustered outputs. Therefore, this dissertation adjusted the target of the cost function from identity matrix to diagonal matrix to train the embedding to become orthogonal, as shown below:

$$P_{orthog} = \left\|V^TV - diag(V^TV)\right\|_F^2. \tag{4.9}$$

This modification allows the embedding to maintain the diagonal term and only train the off-diagonal term to become 0. Therefore, the overall training cost function of the

proposed system is as follows:

$$C_Y(V) = (\left\|V^TV\right\|_F^2 - 2\left\|V^TY\right\|_F^2 + \left\|Y^TY\right\|_F^2) \\ +(\left\|V^TV - diag(V^TV)\right\|_F^2), \tag{4.10}$$

which helps to learn the embeddings to match the target affinity matrix and be orthogonal. The whole system architecture is shown in Figure 4.4, and a similar idea of the penalization term has been introduced in a self-attentive word embedding task [32] without being used for any other types of signals.

4.3.3 Visualization of embedding covariance matrix

Existing speech separation techniques, *i.e.* deep clustering [14], DAN [16], PIT [33, 34], simply decompose the label indicator through the projection without any established standards. Therefore, it is difficult to figure out the degree of decomposition. The penalization term in the proposed method is a criterion that enables to sparsely project the label indicator to the embedding matrix, and it can be verified with Figure 4.5. This figure indicates the covariance matrix among embedding dimensions of 40, and it compares that of conventional deep clustering, proposed method with orthonormal embedding, and proposed method with orthogonal embedding. Figure 4.5(a) demonstrates that the conventional deep clustering method has some focused data on specific embedding dimensions; whereas, the proposed method equally spreads the data throughout all dimensions as shown in Figure 4.5(b) and (c). Observing the covariance matrix between Figure 4.5(b) and (c), orthogonal embedding has more power focused in the center compared to the orthonormal embedding. This indicates that orthogonal embedding uses more dimensions than orthonormal embedding.

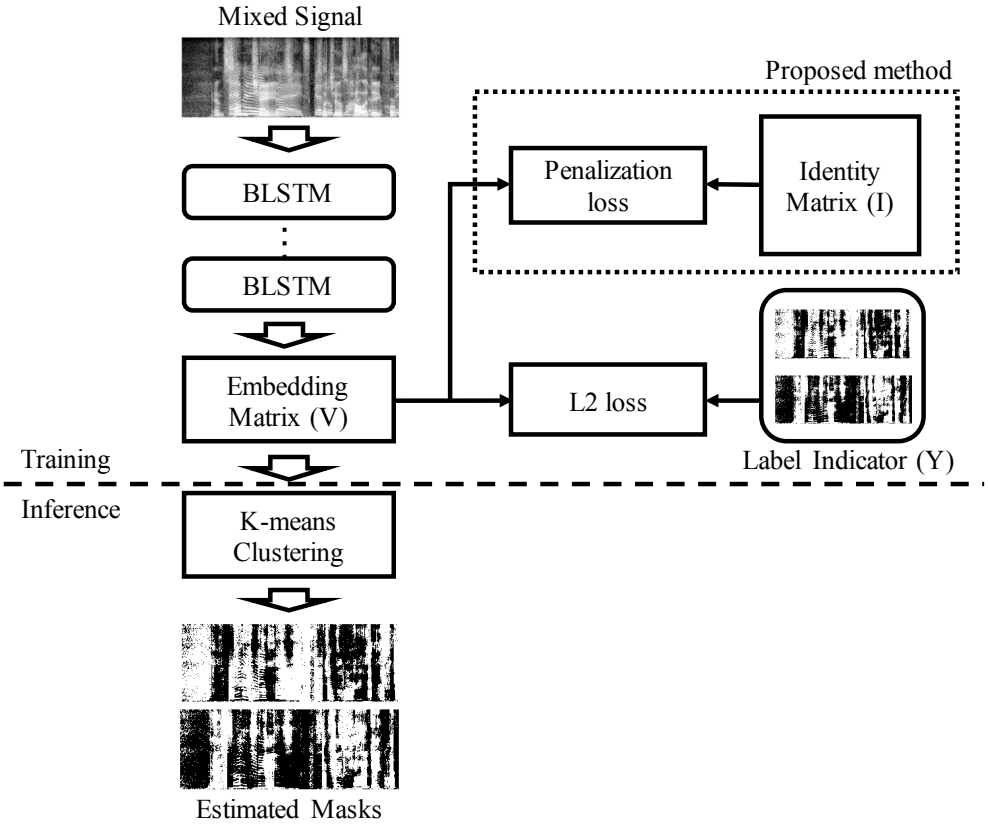


Figure 4.4: Proposed system architecture

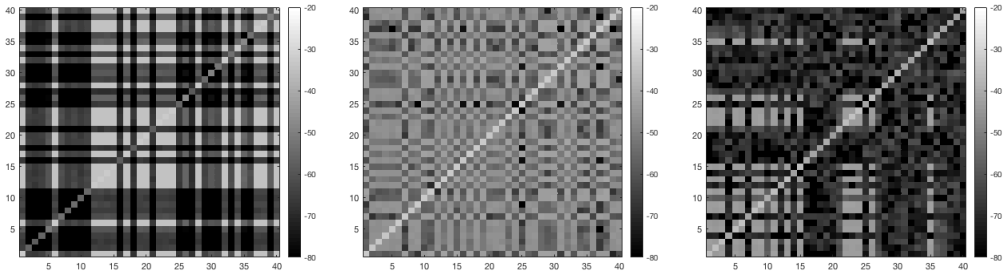


Figure 4.5: (a) Covariance matrix for conventional deep clustering method, (b) covariance matrix for proposed method with orthonormal embedding, (c) covariance matrix for proposed method with orthogonal embedding. The embedding dimension is set to 40.

Chapter 5

Performance Evaluation

In this chapter, the experimental setup of the proposed method is introduced with explanation of the training and inference procedure. As the experiments are proceeded, the results are analyzed with evaluation metric of signal-to-distortion ratio (SDR) followed by some discussions in regard to the result.

5.1 Experimental setup

Deep clustering was evaluated on speaker-independent mixed signals. Here, the mixtures of two speakers were trained with interfering signals of 2 to 15 dB signal to interference ratio (SIR) and evaluated with SIR from 3 to 15 dB in multiple of 3.

Wall Street Journal (WSJ) corpus [35] is used to make speech mixtures using WSJ0. Consequently, 10000 utterances of training set and 5000 utterances of both validation and evaluation sets are generated by randomly selecting the speakers with the same proportion of females and males. Especially for the evaluation set, a set of 1000 mixed utterances of 2 gender cases, *i.e.* same gender and different gender mixtures, is used

with interfering ratio of 3, 6, 9, 12, and 15 dB. Since SIR is evenly distributed in the evaluation set, it becomes easier to analyze the mixtures among gender and SIR.

Sampling frequency of all data is set to 16 kHz, and the input feature X is log spectral magnitude using STFT with 512 samples and 256 samples window shift of hanning window. Different from the conventional deep clustering network, the proposed network uses whole utterance for the input instead of 100 frames, and the deep neural network is implemented with Tensorflow libraries [36].

5.1.1 Training procedure

In order to train the proposed algorithm with a deep learning network, the ideal binary mask (IBM) is used as the target, *i.e.* power dominant bin is set to '1' and '0' for others in each time-frequency bin. The network structure contains two bi-directional long short-term memory (BLSTM) layers followed by one fully connected layer. Each BLSTM layer consists 512 hidden cells and the fully connected layer with the size of the embedding dimension K. Adam optimizer [37] with the learning rate of 10^{-4} is used for training, and hyperbolic tangent is used for both activation and output function of the network. The training criteria are L2 loss between affinity matrices and the additional penalization term, and the dropout rate is set to 0.5 for regularization. Also, several network models using embedding dimensions of 20, 40, and 80 are generated through the experiment.

5.1.2 Speech separation procedure

At inference stage, speech separation is performed by K-means clustering. The output mask is generated by clustering the embeddings V, *i.e.* the output from the proposed model for each utterance. The number of clusters is set to 2 due to the assumption to have 2 speaker mixtures in this experiment .[1)] Among various types of masks, binary masks with the clustered outputs are generated to apply them to the mixture and obtain separated sources. Several types of speech separation, *i.e.* the conventional deep clustering model, deep attractor network, and the proposed models, were evaluated, and the results were compared. For choosing the target speech from the clustered output, the speech with dominant power level was chosen as the target speech and the other as the interfering speech. For all experiments, averaged signal-to-distortion ratio (SDR) was used as the evaluation metric with *mir_eval* library [38], and improved normalized projection alignment (NPA) and relative error rate were analyzed to evaluate the estimated masks.

5.2 Results and discussion

The proposed results were analyzed in five cases: SDR vs. network, SDR vs. dimension, SDR vs. SIR, SDR vs. gender, and mask performance. Observing the separation performance with various networks with embedding dimension of 40 in Table 5.1, the proposed method with orthogonal embeddings showed the best performance. Even though DANet indicated high improvement in separation performance in [16], Table 5.1 showed

[1)] It is possible to extend the system to 3 or more speaker mixtures.

Table 5.1: SDR vs. Network

Network	**SDR (dB)**
DC	11.306
DANet	11.395
DANet$_{orthon}$	11.486
Prop$_{orthon}$	11.437
Prop$_{orthog}$	**11.545**

small increase in the performance, as DANet was organized with basic settings, *i.e.* attractor defined with K-means clustered outputs. In this dissertation, basic DANet was implemented, and the penalization term was added to the network to see its effect on the embeddings. Consequently, applying the penalization term showed higher performance; however, it did not give much influence on the separation. After knowing the aspects of different network performances, other analyzed cases were compared with conventional deep clustering and proposed methods.

With the second case, the effect of the penalization term among different embedding dimensions was analyzed. As shown in the average performance in Table 5.2, proposed method with orthogonal embeddings obtained better result with 20 embedding dimension than the baseline deep clustering method with 80 embedding dimension. Also, as the embedding dimension increased, the improvement rate between proposed method and the baseline increased, meaning that the signals decompose better as embedding dimension increases. From this experiment, the assignment of less correlation to the embeddings by the addition of penalization term to deep clustering method was verified

with improvements in performance. That is because less correlated embeddings result in better decomposition of spectral bins and lead to preferable clustering.

With the third case, the effect of penalization term among different SIRs was analyzed. Table 5.2 also shows the absolute SDR in embedding dimensions of 20, 40, and 80 for various SIRs. As SIR increased, the effect of penalization term increased; however, there were some points in need of attention. In embedding dimension of 20, proposed method with orthonormal embeddings showed degraded performance in low SIRs compared to the conventional deep clustering method, meaning that the embedding dimensions below 20 may not be enough to generate orthonormal embeddings for low SIRs. Forcing to give orthonormality to embeddings can break the correlation when it is actually needed; therefore, it could give more distortion to the signal instead. However, observing the performance for the proposed method with orthogonal embeddings, it showed improved performance in all SIRs indicating that forming orthogonal embedding allows the network to decompose spectral bins to every dimensions.

With the fourth case, the result between genders was analyzed, and it verified that the penalization term works better on mixtures with different genders than same genders as shown in Table 5.3. With embedding dimension of 20, the same gender mixtures showed degradation in performance with penalization term. This confirmed that the embedding dimension below 20 is not enough to increase the effect of penalization term. Also, observation was made that the penalization term works better with embeddings already containing some independency, such as mixture signals with mixed gender. Synthetically, proposed method with orthogonal embeddings showed better performance than with orthonormal embeddings, indicating it is more difficult to properly train the em-

bedding when diagonal terms need to be close to ’1’.

Table 5.2: SDR vs. SIR

		SIR (dB)					
Dimension	**Method**	**3**	**6**	**9**	**12**	**15**	**Avg.**
80	DC	6.039	8.977	11.404	13.887	16.391	**11.339**
	Prop_{orthon}	6.152	9.247	11.880	14.074	16.442	**11.561**
	Prop_{orthog}	6.240	9.348	12.053	14.358	16.626	**11.725**
40	DC	5.816	8.819	11.350	14.062	16.483	**11.306**
	Prop_{orthon}	6.093	8.839	11.623	14.111	16.520	**11.391**
	Prop_{orthog}	6.339	9.299	11.889	14.140	16.058	**11.545**
20	DC	6.138	8.961	11.288	13.809	16.194	**11.278**
	Prop_{orthon}	5.666	8.624	11.398	14.008	16.478	**11.235**
	Prop_{orthog}	6.244	9.086	11.422	14.007	16.553	**11.462**

Table 5.3: SDR vs. Gender

	Same Gender			Mixed Gender		
Dimension	**DC**	**Prop**$_{orthon}$	**Prop**$_{orthog}$	**DC**	**Prop**$_{orthon}$	**Prop**$_{orthog}$
80	10.414	10.812	11.044	12.265	12.310	12.406
40	10.836	10.561	10.733	11.776	12.313	12.357
20	11.292	10.392	10.830	11.264	12.078	12.095

Also, the improvement in permutation problem with the proposed method has been verified with estimated mask analysis with improved normalized projection alignment (NPA) metric [39] and relative error rate, as shown in Table 5.4. Improved NPA shows the improvement in mask formation between deep clustering and the proposed methods, and the relative error rate compares the improved bin prediction error between the target and estimated masks. As the undesired segments in the separated outputs indicate the permutation problem, the quality of the estimated masks is associated with the effective separation of mixed speech. With high SIR mixtures, conventional deep clustering method already obtains high performance, meaning it has less permutation problem. On the other hand, low SIR mixtures haVe difficulty in forming an optimal mask in speech separation; therefore, they are in need of more improvements. Since the embedding dimension is set to 80, proposed method with orthogonal embeddings showed the better performance in both improved NPA and relative error rate than the proposed method with orthonormal embeddings.

5.2.1 Spectrogram samples of proposed algorithm

Having the mixture spectrogram as an input of the proposed system, clustered outputs are reconstructed to obtain spectrograms of the separated sources. Figure 5.1 is the output of mixed gender mixture signal with the SIR of 3 dB, and Figure 5.2 is the output of same gender mixture signal, also with the SIR of 3 dB. By observing both figures (c) and (d), proposed method with orthonormal method shows less distortion in separation than the conventional deep clustering method in some portions. Moreover, the modification of the penalization term by targeting orthogonal embedding increased the overall separation

performance by improving the existing permutation problem shown in Figure 5.1(e) and Figure 5.2(e). Therefore, the spectrogram outputs verified the efficacy of proposed method with orthogonal embedding matrix.

Table 5.4: Quality measurements of estimated masks

		SIR (dB)					
Metric	**Method**	**3**	**6**	**9**	**12**	**15**	**Avg.**
Improved NPA (dB)	Prop_{orthon}	6.258	8.476	4.046	4.957	5.967	**5.941**
	Prop_{orthog}	11.087	13.392	8.996	12.078	5.757	**10.262**
Relative Error Rate	Prop_{orthon}	-0.371	-0.388	-0.142	-0.088	0.137	**-0.170**
	Prop_{orthog}	-0.554	-0.422	-0.422	-0.440	-0.237	**-0.450**

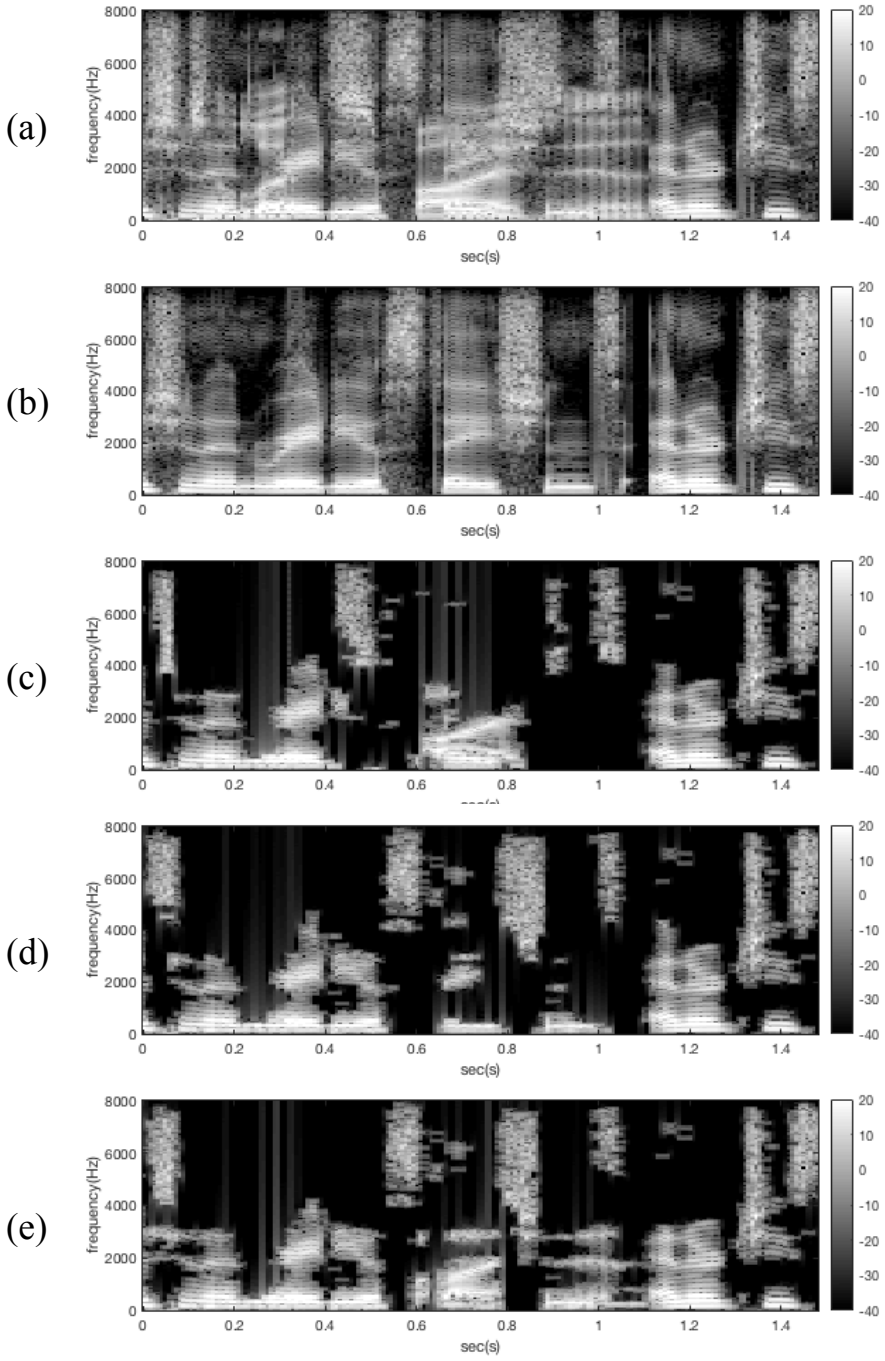


Figure 5.1: Spectrograms of separated sources with different gender mixture. (a) Mixture signal, (b) target signal, (c) estimated output of deep clustering, (d) estimated output of proposed method with orthonormal embeddings, and (e) estimated output of proposed method with orthogonal embeddings.

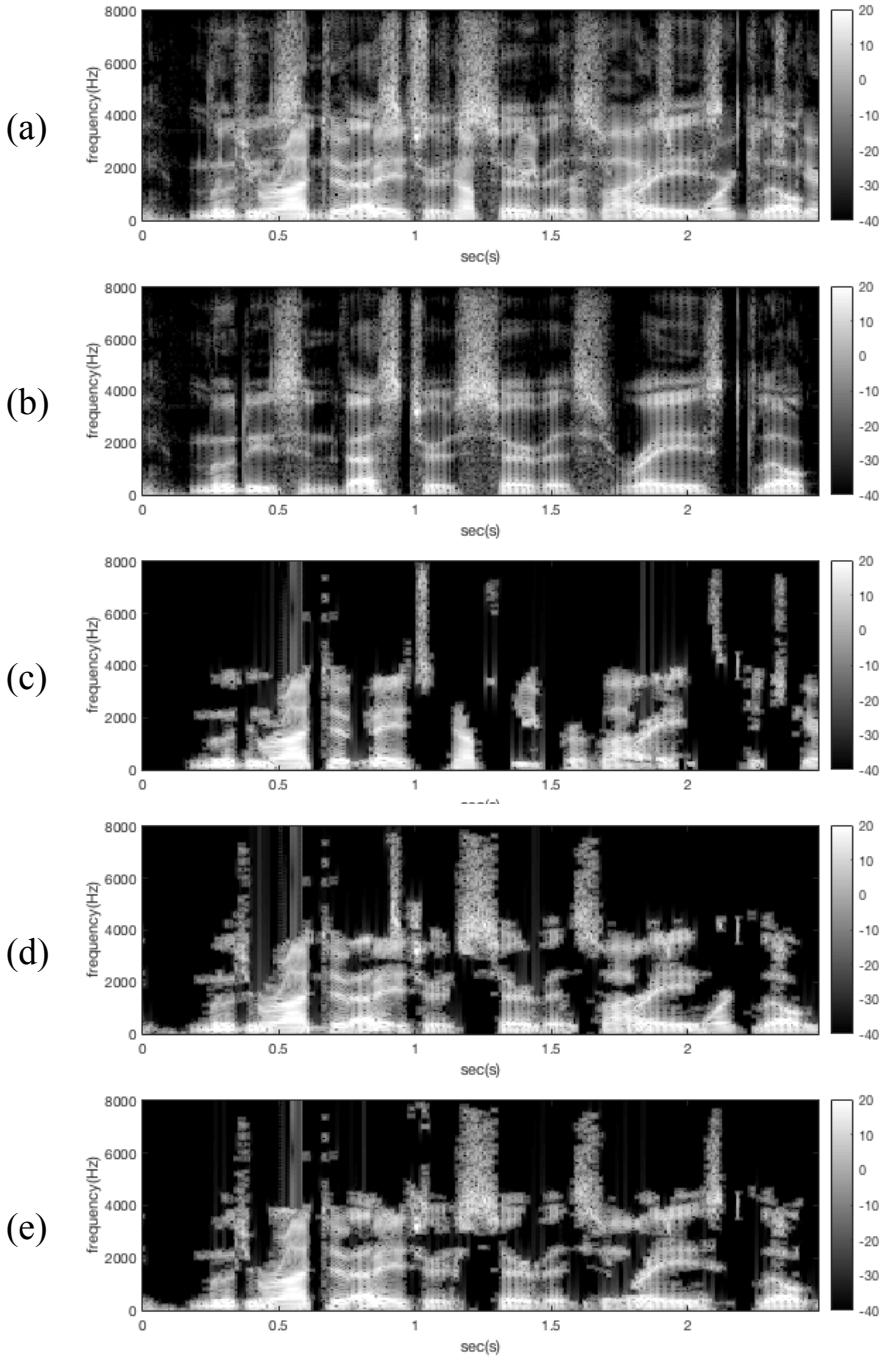


Figure 5.2: Spectrograms of separated sources with same gender mixture. (a) Mixture signal, (b) target signal, (c) estimated output of deep clustering, (d) estimated output of proposed method with orthonormal embeddings, and (e) estimated output of proposed method with orthogonal embeddings.

Chapter 6

Conclusion

The objective of this dissertation is to analyze and enhance the embeddings used in speech separation and consequentially supplement the conventional deep network speech separation methods. In this dissertation, the efficiency of the proposed method was observed through the comparison with the conventional deep clustering and deep attractor network.

Analyzing the covariance matrix of the embedding output of both conventional and proposed methods, it was confirmed that penalization term sparsely projects the label indicator to all the dimensions in the embedding matrix. Also through diverse experiments, it was verified that the performance of the proposed method is maximized when the embedding already contains some independence assumption between speakers in the mixture, *i.e.* high embedding dimension, high SIR, and different gender. Moreover, the permutation problem in the conventional deep clustering method was improved, verified with the evaluation metric for the estimated masks.

After observing the efficacy of penalization term on embeddings, this dissertation

further extended the studies by implementing this regularization term on other embedding-based speech separation methods, *e.g.* deep attractor network [16]. Also, the analysis on the potentials and limitations of the proposed method was done to stabilize the penalization term and optimize its effect on the embeddings by modifying the term to target orthonormal matrix to orthogonal matrix. This modification showed better performance in separation and stable results in all conditions.

References

[1] N. Mesgarani and E. F. Chang, "Selective cortical representation of attended speaker in multi-talker speech perception," *Nature*, vol. 485, pp. 233–236, 2012.

[2] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5745–5749, March 2016.

[3] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, and et al. in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 173–182, JMLR.org, 2016.

[4] Y.-m. Qian, C. Weng, X.-k. Chang, S. Wang, and D. Yu, "Past review, current progress, and challenges ahead on the cocktail party problem," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, pp. 40–63, Jan 2018.

[5] E. Colin Cherry, "Some experiments on the recognition of speech with one and with two ears," *Journal of The Acoustical Society of America*, vol. 25, 9 1953.

[6] M. Cooke, *Modelling Auditory Processing and Organisation*. New York, NY, USA: Cambridge University Press, 1993.

[7] K. Hu and D. Wang, "An unsupervised approach to cochannel speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 122–131, 2013.

[8] G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1135–1150, Sept. 2004.

[9] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1 – 12, 02 2007.

[10] U. Şimşekli, J. L. Roux, and J. R. Hershey, "Non-negative source-filter dynamical system for speech enhancement," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6206–6210, May 2014.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, (Cambridge, MA, USA), pp. 535–541, MIT Press, 2000.

[12] S. Araki, H. Sawada, and S. Makino, "Blind speech separation in a meeting situation with maximum snr beamformers," in *2007 IEEE International Conference*

on Acoustics, Speech and Signal Processing - ICASSP '07, vol. 1, pp. I–41–I–44, April 2007.

[13] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[14] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," 2016.

[15] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *INTERSPEECH*, 2016.

[16] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," 2017.

[17] R. Schluter and H. Ney, "Using phase spectrum information for improved speech recognition performance," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 1, pp. 133–136 vol.1, May 2001.

[18] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, pp. 443–445, April 1985.

[19] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech Language*, vol. 8, no. 4, pp. 297 – 336, 1994.

[20] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[21] D. P. W. Ellis, *Prediction-driven Computational Auditory Scene Analysis*. PhD thesis, Cambridge, MA, USA, 1996. AAI0597425.

[22] A. S. Bregman, *Auditory scene analysis: the perceptual organization of sound*. The MIT Press, 2001.

[23] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717 – 727, 2000.

[24] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. Boston, MA, USA: PWS Publishing Co., 1996.

[25] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[27] Y. Chauvin and D. E. Rumelhart, eds., *Backpropagation: Theory, Architectures, and Applications*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1995.

[28] S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Advances in Neural Information Processing Systems 1* (D. S. Touretzky, ed.), pp. 177–185, Morgan-Kaufmann, 1989.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint*, vol. arXiv, 07 2012.

[30] T. Moon, H. Choi, H. Lee, and I. Song, "Rnndrop: A novel dropout for rnns in asr," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 65–70, Dec 2015.

[31] L. van der Maaten and G. E. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[32] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," 2017.

[33] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," 2017.

[34] M. Kolbaek, D. Yu, Z.-H. Tan, and J. 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, pp. 1901–1913, Oct. 2017.

[35] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, (Stroudsburg, PA, USA), pp. 357–362, Association for Computational Linguistics, 1992.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015.

[38] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "Mir_eval: A transparent implementation of common mir metrics," in *ISMIR*, 2014.

[39] D. Schmid and G. Enzner, "Cross-relation-based blind simo identifiability in the presence of near-common zeros and noise," *IEEE Transactions on Signal Processing*, vol. 60, pp. 60–72, Jan 2012.