SEPNET: A DEEP SEPARATION MATRIX PREDICTION NETWORK FOR MULTICHANNEL AUDIO SOURCE SEPARATION

Shota Inoue[1], *Hirokazu Kameoka*[2], *Li Li*[1], *Shoji Makino*[1]

[1]University of Tsukuba, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Japan

`s.inoue@mmlab.cs.tsukuba.ac.jp, hirokazu.kameoka.uh@hco.ntt.co.jp`

ABSTRACT

In this paper, we propose *SepNet*, a deep neural network (DNN) designed to predict separation matrices from multichannel observations. One well-known approach to blind source separation (BSS) involves independent component analysis (ICA). A recently developed method called independent low-rank matrix analysis (ILRMA) is one of its powerful variants. These methods allow the estimation of separation matrices based on deterministic iterative algorithms. Specifically, ILRMA is designed to update the separation matrix according to an update rule derived based on the majorization-minimization principle. Although ILRMA performs reasonably well under some conditions, there is still room for improvement in terms of both separation accuracy and computation time, especially for large-scale microphone arrays. The existence of a deterministic iterative algorithm that can find one of the stationary points of the BSS problem implies that a DNN can also play that role if designed and trained properly. Motivated by this, we propose introducing a DNN that learns to convert a predefined input (e.g., an identity matrix) into a true separation matrix in accordance with a multichannel observation. To enable it to find one of the multiple solutions corresponding to different permutations of the source indices, we further propose adopting a permutation invariant training strategy to train the network. By using a fully convolutional architecture, we can design the network so that the forward propagation can be computed efficiently. The experimental results revealed that SepNet was able to find separation matrices faster and with better separation accuracy than ILRMA for mixtures of two sources.

Index Terms— Blind source separation, microphone array, deep neural network, permutation invariant training

1. INTRODUCTION

Blind source separation (BSS) is a technique for extracting individual source signals from mixture signals recorded by a microphone array without any prior information about the source signals and the transfer characteristics between the sources and microphones. One widely used approach for the determined BSS problem is independent component analysis (ICA) [1], which achieves source separation by finding a separation matrix that makes the separated signals as statistically independent as possible. Many ICA-based methods are formulated in the frequency domain. Frequency-domain methods enable fast implementations compared to methods formulated in the time domain. In addition, they allow us to utilize various models for the time-frequency representations of source signals and array responses to find clues for separation matrix estimation. For example, independent vector analysis (IVA) [2–4] is a frequency-domain method that solves frequency-wise source separation and permutation alignment simultaneously by assuming that the magnitudes of the frequency components originating from the same source tend to vary coherently over time. Determined multichannel non-negative matrix factorization [5], later called independent low-rank matrix analysis (ILRMA) [6, 7], is another relatively recent example that adopts the non-negative matrix factorization (NMF) concept to model the power spectrogram of each source. Recently, several attempts have also been made to combine deep neural networks (DNNs) with the ICA-based methods or other methods for further improvement [8–15]. Their aim is to improve the representation power of the source spectrogram model employed in multichannel NMF [5, 16, 17] or ILRMA so that better separation matrices can be found.

All the methods above involve iterative procedures for estimating the separation matrix. A fast update rule for the separation matrix has been proposed in [4]. Although this update rule allows the separation matrix to converge with a small number of iterations, the computational cost per iteration can be expensive especially for large-scale microphone arrays. This can be problematic, for example, when implementing real-time BSS systems. To address this, a faster update rule has recently been proposed [18]. While these update rules are derived manually, in this paper, we take a learning-based approach with the aim of automatically obtaining even faster and more accurate update rules using a DNN. The idea is similar in spirit to the concept of deep unfolding [19] in that each layer of a DNN is interpreted as a single update step in an iterative algorithm.

The existence of a deterministic iterative algorithm that leads to one of the stationary points of the BSS problem motivates us to expect that a DNN, if designed and trained properly, can also play that role. Namely, we can think of a DNN that converts a predefined input (corresponding to an initial point of an iterative algorithm) into the true separation matrix in accordance with a multichannel observation. However, implementing such a DNN is not necessarily straightforward, since the BSS problem is an ill-posed one that can have multiple solutions corresponding to different permutations of the source indices. A deterministic BSS algorithm can find one of these solutions, but in most cases it cannot predict in advance which permutation it will end up with. The problem is that the resulting permutation varies depending on an initial point of the iterative algorithm and observed signals. Thus, naively using the source signals arranged in some fixed prespecified order as a regression target for model training would result in a useless model, namely one that produces a separation matrix that does not achieve separation at all. To allow the DNN to flexibly find one of the desired solutions in

This work was supported by JST CREST Grant Number JPMJCR19A3 and JSPS KAKENHI Grant Number 19H04131.

978-1-7281-7605-5/21/$31.00 ©2021 IEEE

accordance with an observation, we further propose adopting a permutation invariant training (PIT) strategy [20] to train the network. On top of this, we use a fully convolutional architecture to design the network so that the forward propagation can be computed efficiently. We call this model *SepNet*, which stands for the *Se*paration matrix *p*rediction *Net*work.

The rest of this paper is structured as follows. Section 2 formulates the BSS problem and briefly reviews ILRMA, Section 3 describes the details of the architecture of SepNet and discusses the training process, and Section 4 presents the experimental results.

2. PROBLEM FORMULATION

2.1. Determined BSS

We consider a determined situation where J source signals are observed by I microphones ($J = I$). Let $x_i(f, n)$ and $s_j(f, n)$ denote the short-time Fourier transform (STFT) coefficients of the signal observed at the ith microphone and the jth source signal, where f and n are the frequency and time indices, respectively. Under a determined situation, we can use a separation system of the form

$$\mathbf{s}(f, n) = \mathbf{W}^{\mathsf{H}}(f)\mathbf{x}(f, n), \tag{1}$$

$$\mathbf{W}(f) = [\mathbf{w}_1(f), \ldots, \mathbf{w}_J(f)] \in \mathbb{C}^{I\times J}, \tag{2}$$

to describe the relationship between the observed signals $\mathbf{x}(f, n) = [x_1(f, n), \ldots, x_I(f, n)]^{\mathsf{T}} \in \mathbb{C}^I$ and sources $\mathbf{s}(f, n) = [s_1(f, n), \ldots, s_J(f, n)]^{\mathsf{T}} \in \mathbb{C}^J$, where $\mathbf{W}(f)$ is called the separation matrix and $(\cdot)^{\mathsf{H}}$ denotes Hermitian transpose. The aim here is to estimate $\mathcal{W} = \{\mathbf{w}_j(f)\}_{j,f}$ from the observations $\mathcal{X} = \{x_i(f, n)\}_{i,f,n}$.

2.2. ILRMA

If we assume each source j to independently follow the local Gaussian model [21, 22] $s_j(f, n) \sim \mathcal{N}_{\mathbb{C}}(s_j(f, n)|0, v_j(f, n))$, where $v_j(f, n)$ denotes the power spectral density of the (f, n)th time-frequency element, the negative log-likelihood of $\mathcal{V} = \{v_j(f, n)\}_{f,n,j}$ and $\mathcal{W} = \{\mathbf{w}_j(f)\}_{j,f}$ given the observed mixture signals $\mathcal{X} = \{x_i(f, n)\}_{i,f,n}$ is given as

$$\mathcal{L}(\mathcal{V}, \mathcal{W}) \overset{c}{=} -2N \log\left|\det \mathbf{W}^{\mathsf{H}}(f)\right| + \sum_{f,n,j}\left(\log v_j(f, n) + \frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f, n)|^2}{v_j(f, n)}\right), \tag{3}$$

where $=^c$ denotes equality up to constant terms.

ILRMA [5–7] is a BSS method that incorporates the NMF model into (3) by expressing $v_j(f, n)$ as the linear sum of M basis spectra $\mathcal{B} = \{b_{j,m}(f)\}_{j,m,f} \geq 0$ scaled by time-varying magnitudes $\mathcal{H} = \{h_{j,m}(n)\}_{j,m,n} \geq 0$. Namely, $v_j(f, n) = \sum_{m=1}^{M} b_{j,m}(f)h_{j,m}(n)$. The optimization algorithm of ILRMA consists of iteratively updating $\mathcal{W}$, $\mathcal{B}$ and $\mathcal{H}$ so that (3) is ensured to be nondecreasing at each iteration. To update $\mathcal{B}$ and $\mathcal{H}$, we can employ the expectation-maximization (EM) algorithm or the majorization-minimization (MM) algorithm [5–7]. To update $\mathcal{W}$, we can use the natural gradient method or IP. The update rule derived on the basis of the IP method [4] is given as

$$\mathbf{w}_j(f) \leftarrow (\mathbf{W}^{\mathsf{H}}(f)\mathbf{\Sigma}_{x/v_j(f)})^{-1}\mathbf{e}_j, \tag{4}$$

$$\mathbf{w}_j(f) \leftarrow \mathbf{w}_j(f)/\sqrt{\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{\Sigma}_{x/v_j}(f)\mathbf{w}_j(f)}, \tag{5}$$

where $\mathbf{\Sigma}_{x/v_j}(f) = (1/N)\sum_n \mathbf{x}(f, n)\mathbf{x}^{\mathsf{H}}(f, n)/v_j(f, n)$ is a weighted spatial covariance matrix and $\mathbf{e}_j$ denotes the j-th column of the $I \times I$ identity matrix. The ILRMA algorithm is guaranteed to converge to a stationary point of (3) and experimentally shown to converge quickly with a small number of iterations. As (4) shows, each iteration of the ILRMA algorithm includes a matrix inversion, which becomes computationally expensive especially when J is large.

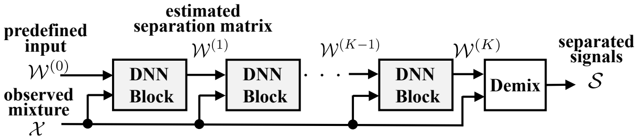


Fig. 1: Entire architecture of proposed method for estimating separation matrix from predefined input and observed mixture.

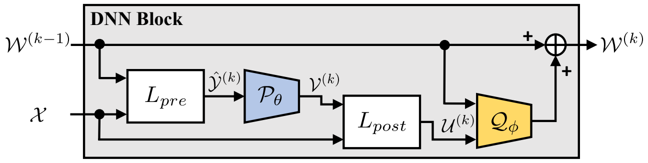


Fig. 2: Diagram of building blocks (DNN Blocks). Each building block consists of two layers with no learnable parameters (L_{pre}, L_{post}) and two layers with learnable parameters ($\mathcal{P}_\theta$, $\mathcal{Q}_\phi$).

3. SEPNET

To obtain a fast and accurate separation matrix estimator, we propose the idea of using an appropriately sized DNN as an alternative to an iterative algorithm for finding $\mathcal{W}$. We call this DNN *SepNet*. To this end, we consider a DNN that takes $\mathcal{X} = \{\mathbf{x}(f, n)\}_{f,n}$ and a set $\mathcal{W}^{(0)}$ of identity matrices as the inputs, and produces the set $\mathcal{W}^{(k)} = \{\mathbf{W}^{(k)}(f)\}_f$ of separation matrix estimates from each block k. The input $\mathcal{W}^{(0)}$ and each block k of this DNN can be thought of as corresponding to the initial point and a single step of a virtual iterative algorithm, respectively.

3.1. Architecture

Fig. 1 and Fig. 2 show the entire architecture of SepNet and the block diagram for each building block. As shown in Fig. 1, the proposed architecture consists of K blocks with the same structure, so the entire network can be viewed as a deep architecture consisting of multiple update processes.

Rather than designing the entire architecture from scratch, we consider it reasonable to design each block by taking inspiration from the update equation shown in Subsec. 2.1. As shown in (4), the update equation in the IP method uses a weighted spatial covariance matrix $\mathbf{\Sigma}_{x/v_j}(f) = (1/N)\sum_n \mathbf{x}(f, n)\mathbf{x}^{\mathsf{H}}(f, n)/v_j(f, n)$ to update each row of $\mathbf{W}^{\mathsf{H}}(f)$, whose weight is determined according to the estimate $v_j(f, n)$ of the power spectral density of the corresponding source. We thus configure each block to consist of four differentiable layers L_{pre}, $\mathcal{P}_\theta$, L_{post}, and $\mathcal{Q}_\phi$ playing the following roles: The first layer, L_{pre}, produces source signal estimates by simply applying the separation matrix estimates produced from the previous block to the observed signals. The second layer, $\mathcal{P}_\theta$, estimates the power spectrogram corresponding to each source signal estimate obtained from L_{pre}. The third layer, L_{post}, computes the weighted spatial covariance matrix using each power spectrogram estimate obtained from $\mathcal{P}_\theta$. The fourth layer, $\mathcal{Q}_\phi$, finally updates the separation matrix estimates using the weighted spatial covariance matrices computed through L_{post}. Here, L_{pre} and L_{post} denote layers with no learnable parameters, whereas $\mathcal{P}_\theta$ and $\mathcal{Q}_\phi$ denote nonlinear layers parameterized by learnable parameters θ and ϕ, respectively. By using $\mathcal{W}^{(k-1)} = \{\mathbf{w}_j^{(k-1)}(f)\}_{j,f}$ to denote the output of the $(k-1)$th block and using $\hat{\mathcal{Y}}^{(k)} = \{\hat{y}_j^{(k)}(f,n)\}_{j,f,n}$, $\mathcal{V}^{(k)} = \{v_j^{(k)}(f,n)\}_{j,f,n}$, and $\mathcal{U}^{(k)} = \{\mathbf{U}_j^{(k)}(f)\}_{j,f}$ to denote the outputs of the intermediate layers in the kth block, the final output $\mathcal{W}^{(k)} = \{\mathbf{w}_j^{(k)}(f)\}_{j,f}$ of the kth block is given as

$$\hat{\mathcal{Y}}^{(k)} = L_{\text{pre}}(\mathcal{W}^{(k-1)}, \mathcal{X}), \tag{6}$$

$$\mathcal{V}^{(k)} = \mathcal{P}_{\theta^{(k)}}(\hat{\mathcal{Y}}^{(k)}), \tag{7}$$

$$\mathcal{U}^{(k)} = L_{\text{post}}(\mathcal{V}^{(k)}), \tag{8}$$

$$\mathcal{W}^{(k)} = \mathcal{W}^{(k-1)} + \mathcal{Q}_{\phi^{(k)}}(\mathcal{W}^{(k-1)}, \mathcal{U}^{(k)}), \tag{9}$$

where each element $\hat{y}_j^{(k)}(f,n)$ of $\hat{\mathcal{Y}}$ and each element $\mathbf{U}_j^{(k)}(f)$ of $\mathcal{U}^{(k)}$ are given respectively as

$$\hat{y}_j^{(k)}(f,n) = |\mathbf{w}_j^{(k-1)\mathsf{H}}(f)\mathbf{x}(f,n)|^2, \tag{10}$$

$$\mathbf{U}_j^{(k)}(f) = \frac{1}{N}\sum_n \frac{\mathbf{x}(f,n)\mathbf{x}^{\mathsf{H}}(f,n)}{v_j^{(k)}(f,n)}. \tag{11}$$

(7) can be seen as a process of refining the power spectrograms of the tentatively separated signals. Note that (9) is designed to have a residual connection, so $\mathcal{Q}_{\phi^{(k)}}$ is responsible for predicting the direction in which $\mathcal{W}^{(k-1)}$ should be updated. As detailed later, we design both $\mathcal{P}_\theta$ and $\mathcal{Q}_\phi$ using fully convolutional architectures.

(7) and (9) assume the use of different parameter sets $\theta^{(k)}$ and $\phi^{(k)}$ for each block. Alternatively, we can think of tying the parameters in all the blocks together (by removing the superscript k from $\theta^{(k)}$ and $\phi^{(k)}$) to reduce the parameters to be learned. To distinguish between these two versions, we call the former the *untied* model and the latter the *tied* model. A comparison of these versions will be made later.

3.2. Training process

The training process is the key to the success of SepNet. Given a training example of the pair $\{\mathcal{X}, \mathcal{S}\}$, one possible option for the training objective would be to use a direct separation error, namely the distance between $\mathbf{w}_j^{(k)\mathsf{H}}(f)\mathbf{x}(f,n)$ and $s_j(f,n)$:

$$E(\theta^{(k)}) = \mathbb{E}_{\mathcal{X},\mathcal{S}}\Big[\sum_j \sum_{f,n} |\mathbf{w}_j^{(k)\mathsf{H}}(f)\mathbf{x}(f,n) - s_j(f,n)|\Big], \tag{12}$$

where $\mathbb{E}_{\mathcal{X},\mathcal{S}}[\cdot]$ denotes the sample mean over all the training examples. However, model training using this objective comes with the permutation problem mentioned earlier. That is, the order of the sources in the target $\mathcal{S}$ may not be the same as the order of the sources in the network output $\mathcal{W}$. To solve this problem, we adopt a PIT strategy [20], which was originally introduced to achieve speaker-independent monaural speech separation. The idea is to first find the best output-target assignment and then minimize the separation error given that assignment. By using $\pi(j) \in \{1, \ldots, J\}$ to denote the source index in the target to which the source index j in the network output is assigned, the PIT objective can be written as

$$E(\theta^{(k)}) = \mathbb{E}_{\mathcal{X},\mathcal{S}}\Big[\min_\pi \sum_j \sum_{f,n} |\mathbf{w}_j^{(k)\mathsf{H}}(f)\mathbf{x}(f,n) - s_{\pi(j)}(f,n)|\Big]. \tag{13}$$

Namely, we evaluate the separation errors for $J!$ possible output-target assignments, from which we find the best assignment π that gives the minimum error for each training example. Our goal is to minimize the mean of the minimum errors computed in this way over all the training examples. Note that only in the $I = J = 2$ case did our method work well even without PIT, by simply sorting the source indices in the target in ascending order of the direction-of-arrivals.

4. EXPERIMENTS

4.1. Experimental settings

We conducted a multichannel speech separation experiment to evaluate our method. ILRMA was chosen as a baseline method for comparison. For the experiment, we created two types of samples of multichannel reverberant mixtures, one being two-channel recordings of two speakers and the other being three-channel recordings of three speakers. To simulate an open-set scenario, we used different speech databases consisting of recordings of completely different speakers to create the training and test sets. Specifically, we used utterances of 18 speakers excerpted from the CMU ARCTIC database [23] for training, and those of four speakers excerpted from the Voice Conversion Challenge (VCC) 2018 dataset [24] for evaluation. The training and test sets consist of 3,000 and 100 samples, respectively. The mixture signals were created from simulated two-channel recordings of two sources and three-channel recordings of three sources, where the room impulse responses were generated by using the image method [25]. The depth, width, and height of the room were set at 4.0, 5.0, and 3.0 m. We created each sample of the training and test sets as follows. First, the reverberation time (T60) was randomly set within the range of 55 to 160 ms. Second, each microphone was randomly placed at least 0.5 m away from the walls of the room. Third, once the positions of the microphone were determined, each speaker was randomly placed at a distance of 0.5 m to 1.0 m from all the microphones so that the angle between each speaker pair with respect to the center of the microphones was at least 20°. All the signals were sampled at 8 kHz. The network architectures for $\mathcal{P}_\theta$ and $\mathcal{Q}_\phi$ are detailed in Fig. 3. We configured $\mathcal{P}_\theta$ to consist of four sub-layers, each of which was designed using a 2D convolution layer (Conv) or a 2D transposed convolution layer (Deconv) with a gated linear unit (GLU) [26]. We configured $\mathcal{Q}_\phi$ to consist of five sub-layers, each of which was designed using a 3D convolution layer (Conv) or a 3D transposed convolution layer (Deconv) with a GLU. In $\mathcal{Q}_\phi$, a complex-valued matrix is treated as a three-way array for convenience of implementation. Namely, when interpreted as an image, the row and column correspond to the height and width, and the real and imaginary parts correspond to first and second channels, respectively. Since the network is fully convolutional, $\mathcal{X}$ is allowed to have an arbitrary length. During training, the utterance was divided into 4.16-s-long segments (128 frames). The tied and untied models were designed to have ten and four blocks, respectively. The STFT was computed using a Hamming window with a length of 64 ms and an overlap of 32 ms. For ILRMA, the basis number M was set to 2, and the algorithm was run for 50 iterations. The initial value of the separation matrix was set at an identity matrix for each frequency bin. All the algorithms were implemented

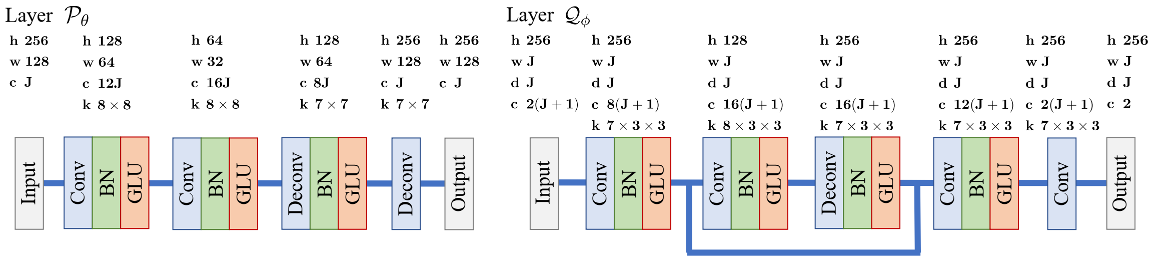
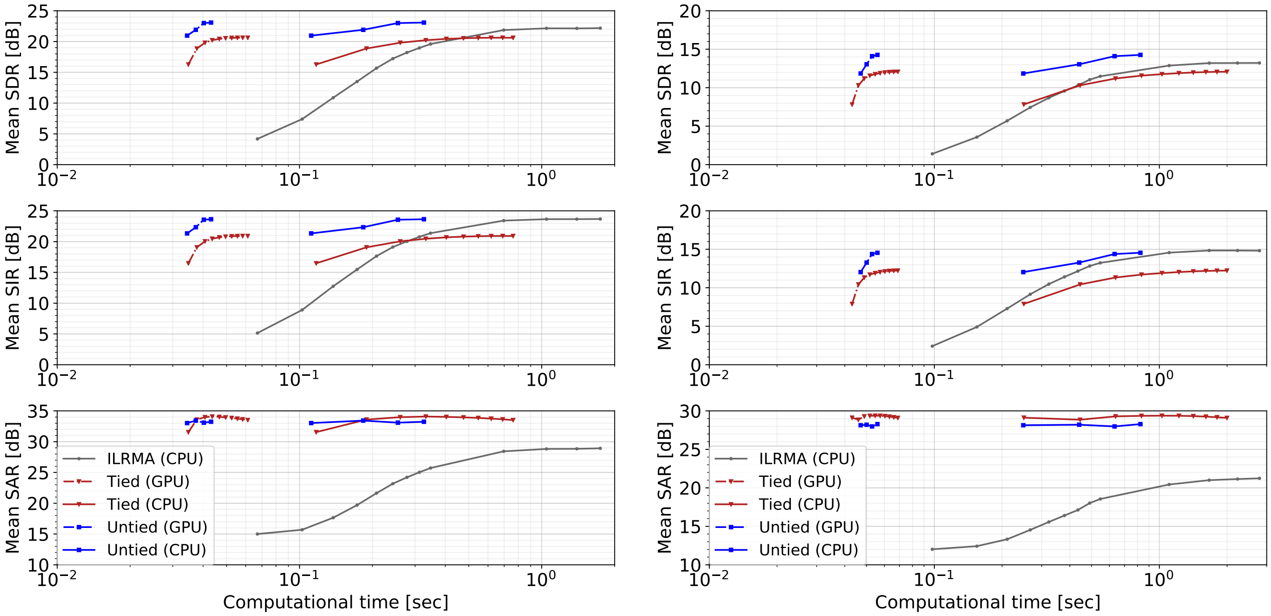


Fig. 3: Detailed network architectures for layer $\mathcal{P}_\theta$ and $\mathcal{Q}_\phi$. "h", "w", "c" and "k" denote the height, width, channel number, and kernel size, respectively. "J" denotes the channel number of input mixture signals. "Conv", "Deconv", "BN", and "GLU" denote 2D or 3D convolution, 2D or 3D deconvolution, batch normalization, and gated linear unit, respectively.



(a) Result for two-speaker separation.

(b) Result for three-speaker separation.

Fig. 4: Average scores of SDR, SIR, and SAR per computational time for ILRMA and both of the proposed methods. The tied model and untied model run ten and four iterations respectively in a CPU or GPU. ILRMA runs 50 iterations in a CPU.

in PyTorch and run on Intel (R) Core i7-7800X CPU@3.50 GHz and GeForce TITAN V GPU.

4.2. Results and Discussion

We computed the average of the signal-to-distortion ratios (SDRs), signal-to-interference ratios (SIRs), and signal-to-artifact ratios (SARs) [27] over the 100 test samples to evaluate the source separation performance and measured the computational time per iteration. Fig. 4 shows graphs of the speech separation accuracy versus run time obtained with the separation matrices estimated at each iteration in ILRMA and at each block in SepNet. The results show that both the tied and untied models could produce good separation matrices even from the first block. The separation matrices produced from the final block in the untied model yielded better separation accuracy than the best accuracy achieved by ILRMA. Moreover, the time it took to reach the final block in the untied model was much shorter than the time ILRMA needed to obtain the best accuracy. Although the tied model performed slightly better than the untied model in terms of the SAR, the untied model performed significantly better in terms of the SDR and SIR. This may indicate that the more flexible the network architecture of SepNet is, the better it performs.

5. CONCLUSION

In this paper, we proposed SepNet, a DNN-based separation matrix predictor for determined frequency-domain BSS. The idea was to let a DNN learn how to find separation matrices from observed signals. For model training, we adopted the PIT strategy to address the permutation mismatch between the order of the sources in the network output and the order of the sources in the regression target. The experimental results revealed that SepNet was able to find separation matrices faster and with better separation accuracy than ILRMA for mixtures of two and three sources.

Although SepNet is advantageous over ILRMA or the IP method in terms of the computational efficiency of the inference process, the training process becomes more challenging as the source/channel number increases. This is because the time complexity of PIT is factorial $\mathcal{O}(J!)$. Hence, developing methods for efficient model training is important.

6. REFERENCES

[1] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4–5, pp. 411–430, 2000.

[2] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ICA to multivariate components," in *Proc. ICA*, 2006, pp. 165–172.

[3] A. Hiroe, "Solution of permutation problem in frequency domain ICA, using multivariate probability density functions," in *Proc. ICA*, 2006, pp. 601–608.

[4] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *Proc. WASPAA*, 2011, pp. 189–192.

[5] H. Kameoka, T. Yoshioka, M. Hamamura, J. Le Roux, and K. Kashino, "Statistical model of speech signals based on composite autoregressive system with application to blind source separation," in *Proc. LVA/ICA*, 2010, pp. 245–253.

[6] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Trans. ASLP*, vol. 24, no. 9, pp. 1626–1641, 2016.

[7] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation with independent low-rank matrix analysis," in *Audio Source Separation*, S. Makino, Ed. Mar. 2018, pp. 125–155, Springer.

[8] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. ASLP*, vol. 24, no. 9, pp. 1652–1664, 2016.

[9] S. Mogami, H. Sumino, D. Kitamura, N. Takamune, S. Takamichi, H. Saruwatari, , and N. Ono, "Independent deeply learned matrix analysis for multichannel audio source separation," in *Proc. EUSIPCO*, 2018.

[10] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Semi-blind source separation with multichannel variational autoencoder," *arXiv preprint arXiv:1808.00892*, Aug. 2018.

[11] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian multichannel speech enhancement with a deep speech prior," in *Proc. APSIPA ASC*, 2018, p. 1233–1239.

[12] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *Proc. ICASSP*, 2019, pp. 101–105.

[13] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Computation*, vol. 31, no. 9, pp. 1891–1914, 2019.

[14] L. Li, H. Kameoka, and S. Makino, "Fast MVAE: Joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," in *Proc. ICASSP*, 2019, pp. 546–550.

[15] S. Inoue, H. Kameoka, L. Li, S. Seki, and S. Makino, "Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder," in *Proc. ICASSP*, 2019, pp. 96–100.

[16] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization inconvolutive mixtures for audio source separation," *IEEE Trans. ASLP*, vol. 18, no. 3, pp. 550–563, 2010.

[17] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Trans. ASLP*, vol. 21, no. 5, pp. 971–982, 2013.

[18] R. Scheibler and N. Ono, "Fast and stable blind source separation with rank-1 updates," in *In Proc. ICASSP*, 2020, pp. 236–240.

[19] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[20] Z.-H. Tan D. Yu, M. Kolbæk and J. Jensen, "Permutation invariant training of deep models for speaker-independent multitalker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

[21] C. Fevotte and J. Cardoso, "Maximum likelihood approach for blind audio source separation using time-frequency gaussian source models," in *In Proc. WASPAA*, 2005, pp. 78–81.

[22] E. Vincent, S. Arberet, and R. Gribonval, "Underdetermined instantaneous audio source separation via local Gaussian modeling," in *Proc. ICA*, 2009, pp. 775–782.

[23] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. SSW*, 2004, pp. 223–224.

[24] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, Apr. 2018.

[25] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[26] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[27] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP*, vol. 14, no. 4, pp. 1462–1469, 2006.