DNN-Based Permutation Solver for Frequency-Domain Independent Component Analysis in Two-Source Mixture Case

Shuhei Yamaji* and Daichi Kitamura*
* National Institute of Technology Kagawa College, Kagawa, Japan

***Abstract*—Frequency-domain independent component analysis (FDICA) is a popular algorithm for multichannel audio source separation. The source components in each frequencies estimated by FDICA must be aligned over all frequencies so that the components of the same source are grouped. This postprocessing of FDICA is the so-called permutation problem. Although various permutation solvers have been proposed, their performances are still limited particularly in a multispeaker separation task in a reverberant environment. To improve the performance of the permutation solver, in this paper, a new data-driven permutation solver based on deep neural networks (DNNs) is presented. In the proposed method, the DNN that predicts whether the input local time-frequency components belong to the same source is trained, and the permutation problem is solved by taking majority decisions of the predicted results.**

I. Introduction

Blind source separation (BSS) is a method of estimating original sources from an observed multichannel mixture signal without knowing a priori information, e.g., the locations of microphones and sources. When the number of microphones is equal to or greater than the number of sources (overdetermined case), independent component analysis (ICA) [1] is a typical approach to solving the BSS problem, which estimates a demixing matrix for the separation.

In general, audio signals are mixed with room reverberation as a convolutive mixture, and simple ICA cannot separate the audio sources. To solve this problem, frequency-domain ICA (FDICA) [2] was proposed. In FDICA, ICA is independently applied in each frequency bin, and the frequency-wise demixing matrices are estimated. However, since ICA cannot determine the order (permutation) of the estimated signals, the frequency-wise components separated by FDICA must be aligned over all frequency bins so that the components of the same source are grouped. This is the so-called *permutation problem*.

A major approach to solving the permutation problem is based on a correlation between time series components of the separated signals in adjacent or local frequency bins [3]. When the positions of microphones are known, the directions of arrivals (DOAs) of the sources can be utilized for taking an alignment of the separated components [4]. The unified permutation solver combining frequency correlation and DOAs was also proposed [5].

This work was partly supported by NVIDIA GPU Grant and JSPS KAKENHI Grant Numbers 19K20306 and 19H01116.

In recent years, BSS algorithms without encountering the permutation problem have been proposed. For example, independent vector analysis (IVA) assumes the co-occurrence of all frequency components of the same source and estimates the frequency-wise demixing matrices avoiding the permutation problem [6], [7]. Independent low-rank matrix analysis (ILRMA) [8], [9] assumes a low-rank time-frequency structure of each source, resulting in a more precise estimation of the permutation-aligned demixing matrices. However, their performances are still limited particularly in a multispeaker separation task. This is because the dominant frequencies of speech signals are substantially overlapped. Moreover, the source models assumed in the correlation-based permutation solver [3], IVA, and ILRMA are not suitable for representing speech time-frequency structures.

In this paper, a new data-driven permutation solver based on deep neural networks (DNNs) is presented. The proposed DNN is trained to predict whether the two input narrowband frequency components (time-varying powers) belong to the same source, where the training data can be prepared by manually shuffling the narrowband frequency components of clean or separated speech spectrograms for all frequency bins. In the test stage, the trained DNN is applied to all frequency bins, and majority decisions of the predicted results along the time frames and the frequency bins are taken to obtain the accurate permutation predictions. The validity of the proposed method is confirmed by experiments of multispeaker audio source separation, where the proposed permutation solver is used after the estimation by simple FDICA.

In the underdetermined case (the number of microphones is less than the number of sources), multichannel audio source separation based on a spatial covariance matrix [10] is the most popular and reliable algorithm. However, this method is also faced with the permutation problem. Since the proposed permutation solver only requires monaural (nonaligned) source estimates, we can apply the proposed method in the underdetermined case without loss of generality.

II. FDICA and Permutation Problem

A. Formulation and FDICA

We define the numbers of audio sources and observed channels (microphones) as N and M, respectively. The source, mixture, nonaligned separated, and aligned separated signals

obtained by short-time Fourier transform (STFT) are respectively represented as

$$\boldsymbol{s}_{i,j} = (s_{i,j,1}, \cdots, s_{i,j,n}, \cdots, s_{i,j,N})^{\mathrm{T}} \in \mathbb{C}^N, \quad (1)$$

$$\boldsymbol{x}_{i,j} = (x_{i,j,1}, \cdots, x_{i,j,m}, \cdots, x_{i,j,M})^{\mathrm{T}} \in \mathbb{C}^M, \quad (2)$$

$$\boldsymbol{y}_{i,j} = (y_{i,j,1}, \cdots, y_{i,j,n}, \cdots, y_{i,j,N})^{\mathrm{T}} \in \mathbb{C}^N, \quad (3)$$

$$\boldsymbol{z}_{i,j} = (z_{i,j,1}, \cdots, z_{i,j,n}, \cdots, z_{i,j,N})^{\mathrm{T}} \in \mathbb{C}^N, \quad (4)$$

where $i = 1, 2, \cdots, I$, $j = 1, 2, \cdots, J$, $n = 1, 2, \cdots, N$, and $m = 1, 2, \cdots, M$ are the indices of frequency bins, time frames, sources, and channels, respectively, and $\cdot^{\mathrm{T}}$ denotes the vector transpose. Also, $s_{i,j,n}$, $x_{i,j,m}$, $y_{i,j,n}$, and $z_{i,j,n}$ are the complex-valued elements of spectrogram matrices $\boldsymbol{S}_n \in \mathbb{C}^{I\times J}$, $\boldsymbol{X}_m \in \mathbb{C}^{I\times J}$, $\boldsymbol{Y}_n \in \mathbb{C}^{I\times J}$, and $\boldsymbol{Z}_n \in \mathbb{C}^{I\times J}$, respectively. Note that the estimated spectrogram $\boldsymbol{Y}_n$ is problematic because the source permutation of narrow-band frequency components $(y_{i,1,n}, \cdots, y_{i,J,n})$ over all frequencies is not aligned, which is called the permutation problem. The permutation-aligned spectrograms $\boldsymbol{Z}_n$ can be obtained by applying a permutation solver to $\boldsymbol{Y}_n$. Thus, the aim of this paper is to estimate $\boldsymbol{Z}_n$ from $\boldsymbol{Y}_n$.

In FDICA, we assume that the mixing system is linear time-invariant in each frequency bin, which can be expressed by the frequency-wise mixing matrix $\boldsymbol{A}_i = (\boldsymbol{a}_{i,1} \ \cdots \ \boldsymbol{a}_{i,N}) \in \mathbb{C}^{M\times N}$. The mixture signal is defined as

$$\boldsymbol{x}_{i,j} = \boldsymbol{A}_i \boldsymbol{s}_{i,j}, \quad (5)$$

where $\boldsymbol{a}_{i,n} \in \mathbb{C}^M$ is the steering vector of the nth source. This mixing model is valid only when the window length in STFT is longer than the length of room reverberation.

When the number of channels is equal to the number of sources ($M = N$) and $\boldsymbol{A}_i$ is a nonsingular matrix, the demixing matrix $\boldsymbol{W}_i = \boldsymbol{A}_i^{-1} = (\boldsymbol{w}_{i,1} \cdots \boldsymbol{w}_{i,N})^{\mathrm{H}} \in \mathbb{C}^{N\times M}$ exists, and the separated signal is defined as

$$\boldsymbol{z}_{i,j} = \boldsymbol{W}_i \boldsymbol{x}_{i,j}, \quad (6)$$

where $\boldsymbol{w}_{i,n} \in \mathbb{C}^M$ is the demixing filter of the nth source and $\cdot^{\mathrm{H}}$ denotes the Hermitian transpose. Therefore, the demixing matrix $\boldsymbol{W}_i$ for all frequency bins must be estimated by FDICA to achieve source separation.

B. Permutation Problem

FDICA cannot estimate the scales (volumes) and orders (permutations) of the separated signals because ICA is based on the statistical independence between sources. Therefore, the uncertainty remains even when the separation is ideally performed as follows:

$$\hat{\boldsymbol{W}}_i = \boldsymbol{D}_i \boldsymbol{P}_i \boldsymbol{W}_i \quad (7)$$

$$= \boldsymbol{D}_i \boldsymbol{P}_i \boldsymbol{A}_i^{-1}, \quad (8)$$

where $\boldsymbol{P}_i$ is a permutation matrix that may replace orders of the row vectors $\boldsymbol{w}_{i,n}$ in $\boldsymbol{W}_i$ and $\boldsymbol{D}_i$ is a diagonal matrix that may vary the scale of $\boldsymbol{w}_{i,n}$ in $\boldsymbol{W}_i$. For this reason, the separated signals estimated by FDICA, $\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N$, have the permutation and scale ambiguities in each frequency bin. The scale ambiguity can easily be recovered by applying the back projection technique [3]. However, estimating the correctly aligned permutation is difficult because this problem includes a combinatorial explosion. The permutation problem is depicted in Fig. 1, where the permutation solver takes an alignment of the separated sources over all frequency bins.

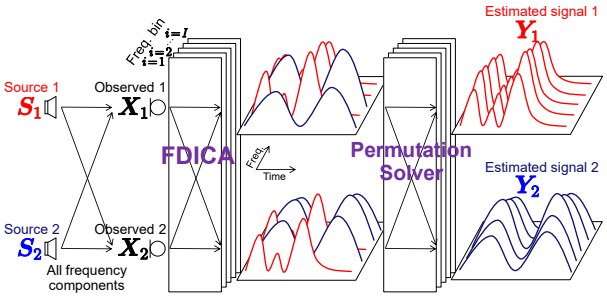

Fig. 1. Permutation problem in FDICA, where $N = 2$.

The permutation-aligned separated signal $\boldsymbol{z}_{i,j}$ is obtained as

$$\boldsymbol{z}_{i,j} = \boldsymbol{P}_i^{-1} \boldsymbol{D}_i^{-1} \boldsymbol{y}_{i,j}$$

$$= \boldsymbol{P}_i^{-1} \boldsymbol{D}_i^{-1} \hat{\boldsymbol{W}}_i \boldsymbol{x}_{i,j} \quad (9)$$

In this paper, we aim to estimate $\boldsymbol{P}_i^{-1}$ in (9) over all frequency bins using a new data-driven permutation solver.

III. Proposed Method

A. Motivation and Strategy

In Ref. [11], the optimal window length in STFT for BSS was experimentally investigated. Fig. 6(b) in Ref. [11] shows that FDICA with the DOA-based permutation solver, IVA, and ILRMA cannot accurately separate speech sources under a reverberant condition ($T_{60} = 470$ ms). However, FDICA with the ideal permutation solver achieves over 10 dB improvement in the signal-to-distortion ratio (SDR) [12], where the ideal permutation solver uses completely separated (oracle) source signals $\boldsymbol{s}_{i,j}$ for estimating the permutation alignment matrix $\boldsymbol{P}_i^{-1}$. This fact implies that the demixing matrix $\hat{\boldsymbol{W}}_i$ itself can be accurately estimated by FDICA even for a reverberant speech mixture, but only the permutation solver fails to estimate $\boldsymbol{P}_i^{-1}$. IVA and ILRMA may also fail to solve only the permutation problem even though it may successfully separate frequency-wise sources. This may be because the source model assumed in IVA or ILRMA is not suitable for speech sources. In fact, the source model in IVA, i.e., the co-occurrence of all frequency components of the same source, is too simplistic to avoid the permutation problem. Also, the source model in ILRMA, i.e., the low-rank time-frequency structure of the same source, does not fit to the power spectrogram of speech signals, which dynamically and continuously changes its spectra.

On the basis of the experimental result in Ref. [11], we can assume that FDICA achieves satisfactory separation in each frequency bin. Thus, in this paper, we only focus on accurately solving the permutation problem and develop a new DNN-based supervised (data-driven) permutation solver. Hereafter,

we only treat a two-source mixture case ($N = 2$). This is because in this study, we focus on only the investigation of the availability of DNN for solving the permutation problem. The algorithm for mixtures with more than two sources is our important future work.

The proposed DNN-based permutation solver is summarized as follows:

- Two frequency-binwise activations (time-varying powers) in the two source spectrograms $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ are input to the DNN.
- The DNN predicts whether the two input binwise activations of each source are the correct permutation as a binary scalar.
- The DNN is applied to scan all frequency bins and time frames of the separated sources $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ estimated by FDICA.
- The conclusive estimate of permutation $\boldsymbol{P}_i^{-1}$ is decided using a majority decision of the predicted results along frequency bins and time frames.

The proposed permutation solver can be interpreted as a supervised and nonlinear extension of the conventional binwise-correlation-based permutation solver [3] because the proposed DNN exploits the relationship between two activations of adjacent or local frequency bins. The training data for the proposed DNN can easily be produced by manually shuffling the clean or separated speech spectrograms to simulate the permutation problem, where the labels of this input data are given by the shuffling result.

B. DNN Input and Output

The input vector for the proposed DNN model is shown in Fig. 2. After applying FDICA to the observed mixture, we obtain the separated spectrograms $\boldsymbol{Y}_n$, which have a permutation problem. From their power spectrograms, $|\boldsymbol{Y}_n|^{\cdot 2}$, two-source ($n = 1$ and 2) and two-frequency-binwise (i and $i + \omega$) short-time activations with length τ are gathered as follows:

$$\boldsymbol{d}_{i,\omega,\gamma} = (\tilde{\boldsymbol{r}}_{i,\gamma}^{\mathrm{T}}, \tilde{\boldsymbol{g}}_{i,\omega,\gamma}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{4\tau\times 1}, \tag{10}$$

$$\tilde{\boldsymbol{r}}_{i,\gamma} = (\boldsymbol{r}_{i,\gamma,1}{}^{\mathrm{T}}, \boldsymbol{r}_{i,\gamma,2}{}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{2\tau\times 1}, \tag{11}$$

$$\boldsymbol{r}_{i,\gamma,n} = (|y_{i,(\gamma-1)\eta+1,n}|^2, |y_{i,(\gamma-1)\eta+2,n}|^2, \cdots, |y_{i,(\gamma-1)\eta+\tau,n}|^2)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{\tau\times 1}, \tag{12}$$

$$\tilde{\boldsymbol{g}}_{i,\omega,\gamma} = (\boldsymbol{g}_{i,\omega,\gamma,1}{}^{\mathrm{T}}, \boldsymbol{g}_{i,\omega,\gamma,2}{}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{2\tau\times 1}, \tag{13}$$

$$\boldsymbol{g}_{i,\omega,\gamma,n} = (|y_{i+\omega,(\gamma-1)\eta+1,n}|^2, |y_{i+\omega,(\gamma-1)\eta+2,n}|^2, \cdots, |y_{i+\omega,(\gamma-1)\eta+\tau,n}|^2)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{\tau\times 1}, \tag{14}$$

where $|\cdot|^{\cdot 2}$ for matrices returns a matrix with element-wise absolute and squared operations, $\omega = -\Omega, -\Omega + 1, \cdots, -1, 0, 1, \cdots, \Omega$ is the index that defines the difference in the number of frequency bins between $\boldsymbol{r}_{i,\gamma,n}$ and $\boldsymbol{g}_{i,\omega,\gamma,n}$, η is the stride length of the short-time activation along the time-frame axis, and $\gamma = 1, 2, \cdots, \Gamma$ is the index of short-time activations. Note that Γ is calculated using the settings of the length of short-time activations τ and the stride length η. The vector $\boldsymbol{r}_{i,\gamma,n}$ corresponds to the short-time activation in the reference frequency bin i, and the vector $\boldsymbol{g}_{i,\omega,\gamma,n}$ is the short-time activation in the adjacent or local frequency bin $i + \omega$ as shown in Fig. 2.

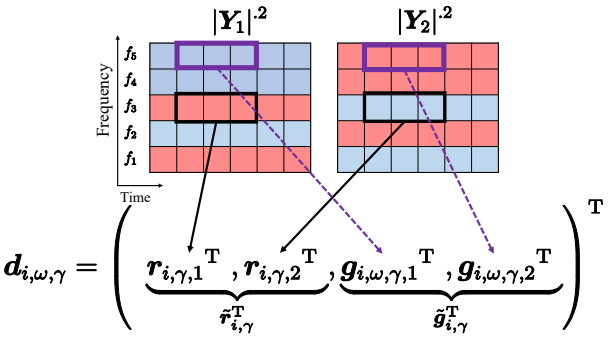

Fig. 2. Input vector of DNN. Matrices $|\boldsymbol{Y}_1|^{\cdot 2}$ and $|\boldsymbol{Y}_2|^{\cdot 2}$ are separated power spectrograms with permutation problem, and red and blue binwise activations (rows of $|\boldsymbol{Y}_1|^{\cdot 2}$ and $|\boldsymbol{Y}_2|^{\cdot 2}$) depict sourcewise components, e.g., red and blue slots respectively correspond to first and second source components.

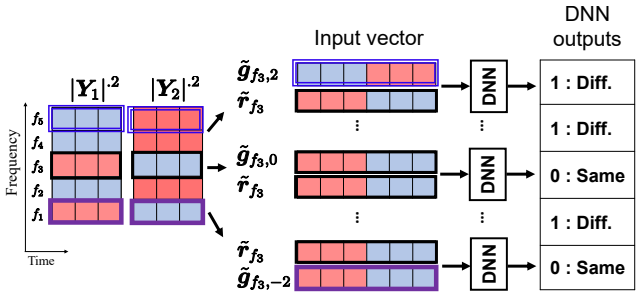

Fig. 3. DNN predictions in subband frequency bins, where $f_1, f_2, \cdots, f_5$ are frequency bins in subband frequency, and index of short-time activations, γ, is omitted for simplicity. Reference frequency bin is $i = f_3$, and adjacent or local frequency bins are $i + \omega = f_1, f_2, \cdots, f_5$, namely, $\Omega = 2$. When source permutation of $\tilde{\boldsymbol{r}}_i$ and $\tilde{\boldsymbol{g}}_{i,\omega}$ is correct, DNN ideally outputs zero as "same." In contrast, when source permutation of $\tilde{\boldsymbol{r}}_i$ and $\tilde{\boldsymbol{g}}_{i,\omega}$ is incorrect, DNN ideally outputs one as "different."

The input vector for DNN is defined as the normalized vector of (10) obtained as

$$\tilde{\boldsymbol{d}}_{i,\omega,\gamma} = \frac{\boldsymbol{d}_{i,\omega,\gamma}}{\|\boldsymbol{d}_{i,\omega,\gamma}\|_2} \in \mathbb{R}_{\geq 0}^{4\tau\times 1}, \tag{15}$$

where $\|\cdot\|_2$ denotes the L_2 norm.

The proposed DNN model is a binary classifier that outputs zero or one. The "zero" output indicates that the permutations of two input short-time activations $(\boldsymbol{r}_{i,\gamma,1}, \boldsymbol{r}_{i,\gamma,2})$ and $(\boldsymbol{g}_{i,\omega,\gamma,1}, \boldsymbol{g}_{i,\omega,\gamma,2})$ are correct, namely, $\boldsymbol{r}_{i,\gamma,1}$ and $\boldsymbol{g}_{i,\omega,\gamma,1}$ are the same source components and $\boldsymbol{r}_{i,\gamma,2}$ and $\boldsymbol{g}_{i,\omega,\gamma,2}$ are also the same source components. In contrast, the "one" output indicates that the permutations of $(\boldsymbol{r}_{i,\gamma,1}, \boldsymbol{r}_{i,\gamma,2})$ and $(\boldsymbol{g}_{i,\omega,\gamma,1}, \boldsymbol{g}_{i,\omega,\gamma,2})$ are incorrect, namely, $\boldsymbol{r}_{i,\gamma,1}$ and $\boldsymbol{g}_{i,\omega,\gamma,1}$ are the different source components and $\boldsymbol{r}_{i,\gamma,2}$ and $\boldsymbol{g}_{i,\omega,\gamma,2}$ are also the different source components. These predicted results are illustrated in Fig. 3. In practice, the result of DNN prediction is not a binary but a soft scalar defined as

$$q_{i,\omega,\gamma} = \mathrm{DNN}\left(\tilde{\boldsymbol{d}}_{i,\omega,\gamma}\right) \in [0, 1], \tag{16}$$

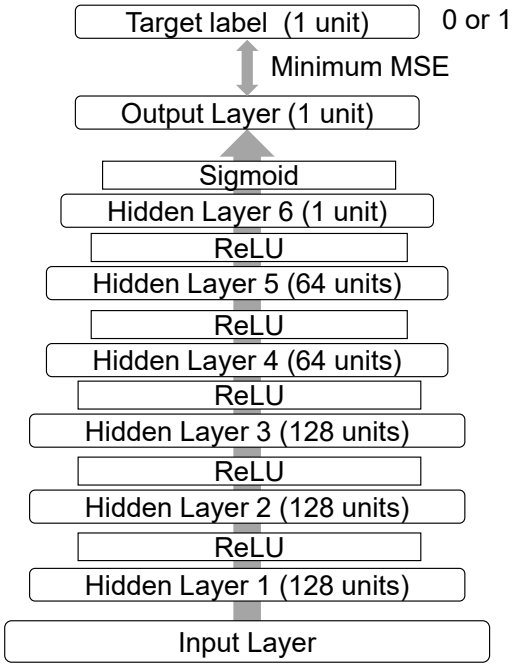

Fig. 4. DNN architecture.

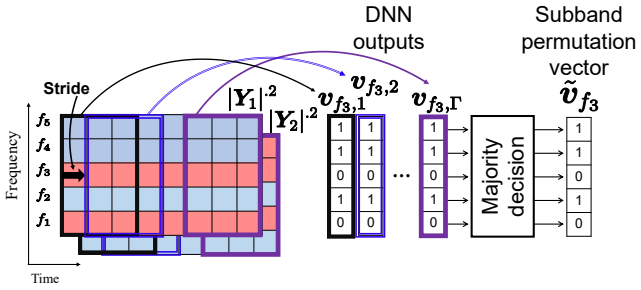

Fig. 5. DNN predictions for all short-time subbands and their majority decisions. Since permutation does not depend on time frames, this majority decision along time-frame axis effectively reduces adverse effect of DNN prediction errors.

which can be interpreted as the reliability of the permutation correctness.

C. DNN Architecture and Training Loss Function

Fig. 4 depicts an architecture of DNN used in the proposed permutation solver. This DNN model has full-connected eight layers (one input layer, six hidden layers, and one output layer). A rectified linear unit (ReLU) is applied after the intermediate hidden layers, and a sigmoid function is applied after the final hidden layer. The loss function is defined as a mean squared error (MSE) between the predicted result $q_{i,\omega,\gamma}$ and its target label.

D. DNN Predictions in Subband Frequency Bins

The examples of DNN prediction are illustrated in Fig. 3, where $f_1, f_2, \cdots, f_5$ are the subband frequency bins, and the index of short-time activations, γ, is omitted for simplicity. In this figure, the reference frequency bin is set to $i = f_3$, and its adjacent or local frequency bins are defined as $i+\omega = f_1, f_2, \cdots, f_5$, namely, $\Omega = 2$. Note that the component in the reference frequency bin f_3 of $\boldsymbol{Y}_1$ corresponds to a red source, but the components in f_2, f_4, and f_5 of $\boldsymbol{Y}_1$ correspond to a blue source. All the combinations of $(\tilde{\boldsymbol{r}}_i, \tilde{\boldsymbol{g}}_{i,\omega})$ are input to the DNN, which are $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,-2}), (\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,-1}), \cdots, (\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,2})$ in the case of Fig. 3. When the permutations of the two input frequency bins (i and $i+\omega$) are correct, the DNN ideally outputs zero (the "same" label). In contrast, when the permutations of the two input frequency bins are incorrect, the DNN ideally outputs one (the "different" label). In Fig. 3, for the combinations $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,-2})$ and $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,0})$, the DNN outputs zero, and for the combinations $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,-1})$, $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,1})$, and $(\tilde{\boldsymbol{r}}_{f_3}, \tilde{\boldsymbol{g}}_{f_3,2})$, the DNN outputs one. As a result, the correctness values of the permutation based on the reference frequency bin i can be estimated as shown on the right side of Fig. 3. This vector is called *(subband) permutation vector*. In practice, the DNN output $q_{i,\omega,\gamma}$ is a value in the range $[0, 1]$. To produce the subband permutation vector, the following thresholding is performed:

$$\tilde{q}_{i,\omega,\gamma} = \mathrm{round}(q_{i,\omega,\gamma}) \in \{0, 1\}, \tag{17}$$

where $\mathrm{round}(\cdot)$ is a rounding operator, and the subband permutation vector can be obtained as

$$\tilde{\boldsymbol{q}}_{i,\gamma} = (\tilde{q}_{i,-\Omega,\gamma}, \tilde{q}_{i,-\Omega+1,\gamma}, \cdots, \tilde{q}_{i,-1,\gamma}, \tilde{q}_{i,0,\gamma}, \tilde{q}_{i,1,\gamma}, \cdots, \tilde{q}_{i,\Omega,\gamma})^{\mathrm{T}} \in \{0, 1\}^{2\Omega+1}. \tag{18}$$

The lengths of the short-time activations $\tilde{\boldsymbol{r}}_{i,\gamma}$ and $\tilde{\boldsymbol{g}}_{i,\omega,\gamma}$ are defined as τ. Since audio signals have a sparse property in the time-Frequency-domain, there are so many time-frequency slots that have almost zero powers. In particular, speech signals have silent intervals due to breath. If the reference activation $\tilde{\boldsymbol{r}}_{i,\gamma}$ is a silent interval of the sources, the prediction of DNN becomes unstable.

To cope with this problem, in the proposed permutation solver, the DNN prediction is multiply applied to all the time frames by shifting the τ-length input vector $\tilde{\boldsymbol{d}}_{i,\omega,\gamma}$ with a η-length stride as shown in Fig. 5. Therefore, we can collect the DNN outputs along the time frames. By taking a majority decision of these DNN outputs, we obtain a more reliable subband permutation vector $\tilde{\boldsymbol{v}}_i$. This process can be described as

$$\boldsymbol{v}_i = \frac{1}{\Gamma} \sum_{\gamma} \tilde{\boldsymbol{q}}_{i,\gamma} \in \{0, 1\}^{2\Omega+1}, \tag{19}$$

$$\tilde{\boldsymbol{v}}_i = \mathrm{round}(\boldsymbol{v}_i) \in \{0, 1\}^{2\Omega+1}, \tag{20}$$

where $\mathrm{round}(\cdot)$ for vectors denotes the element-wise rounding operation. This majority decision is reasonable because the permutation problem does not depend on the time frames, and the adverse effect of the prediction errors is effectively reduced.

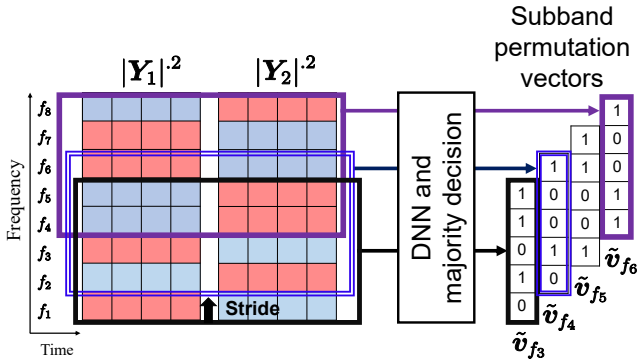

Fig. 6. Estimation of subband permutation vectors in all frequency bins. Subband is shifted with stride length and several overlaps. Similarly to time-frame axis, majority decision is performed for each frequency bin.

E. Estimation of Fullband Permutation Vector

The proposed DNN-based permutation solver consists of the following steps: (a) prediction of the subband permutation vectors in all the frequency bins with subband striding (Sect. III-E1, Fig. 6) and (b) calculation of the fullband permutation vector based on a similarity comparison and a majority decision (Sect. III-E2, Fig. 7).

1) Estimation of Subband Permutation Vectors in All Frequency Bins: The subband permutation vector $\tilde{\boldsymbol{v}}_i$ is estimated in all the frequency bins by shifting the reference frequency bin i (striding subband frequency bins in the range $[i-\Omega, i+\Omega]$) as shown in Fig. 6. Thus, we obtain I subband permutation vectors $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \cdots, \tilde{\boldsymbol{v}}_I$.

Note that the significance of frequency-wise binary scalars in $\tilde{\boldsymbol{v}}_i$ is not identical within $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \cdots, \tilde{\boldsymbol{v}}_I$. This is because the DNN output indicates that the frequency-wise components in $\boldsymbol{Y}_n$ correspond to the "same (zero)" source as the component in the reference frequency bin i or "not (one)" and the source permutation of the reference frequency bin varies depending on i. For example, in Fig. 6, zeros and ones in $\tilde{\boldsymbol{v}}_{f_3}$ respectively indicate "red" and "blue" source components, but zeros and ones in $\tilde{\boldsymbol{v}}_{f_4}$ respectively indicate "blue" and "red" source components (because the reference frequency bins f_3 and f_4 contain red and blue source components, respectively). The alignment of these subband permutation vectors is processed in Sect. III-E2.

2) Reconstruction of Fullband Permutation Vector: From the estimated subband permutation vectors $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \cdots, \tilde{\boldsymbol{v}}_I$, we reconstruct the fullband permutation vector defined as

$$\boldsymbol{u} = (u_1, u_2, \cdots, u_I)^{\mathrm{T}} \in \{0,1\}^I. \tag{21}$$

The reconstruction process of $\boldsymbol{u}$ is depicted in Fig. 7.

As described in Sect. III-E1, the significance of subband permutation vectors $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \cdots, \tilde{\boldsymbol{v}}_I$ is not identical. Therefore, it is necessary to unify the subband permutation vectors over all frequency bins so that the values "zero" and "one" respectively indicate the first and second sources (red and blue sources in Fig. 7).

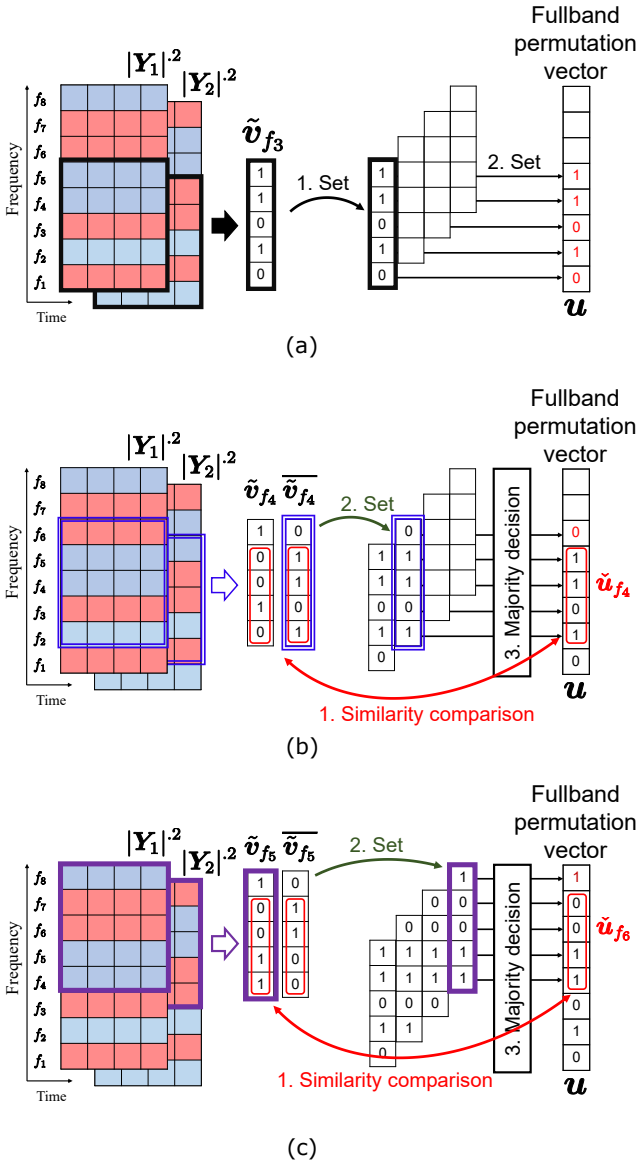

Fig. 7. Reconstruction of fullband permutation label: (a) initialization step, (b) second step, and (c) last step, where similarity comparison is based on MSE. This process is performed to associate DNN predictions (zeros and ones) with each source component.

Fig. 7(a) shows the initial step in the reconstruction of the fullband permutation vector $\boldsymbol{u}$. The subband permutation vector of the lowest frequency subband, $\tilde{\boldsymbol{v}}_{i_\mathrm{s}}$, [$\tilde{\boldsymbol{v}}_{f_3}$ in Fig. 7(a)] is simply set to the corresponding frequency bins in the fullband permutation vector $\boldsymbol{u}$ as shown in Fig. 7(a), where i_s is the index of the lowest reference frequency bin. In Fig. 7(a), since $i_\mathrm{s} = f_3$ and $\Omega = 2$, $u_1, u_2, \cdots, u_5$ are determined by $\tilde{\boldsymbol{v}}_{f_3}$.

Fig. 7(b) shows the following step of Fig. 7(a). In this step, the subband permutation vector adjacent to the lowest frequency subband, $\tilde{\boldsymbol{v}}_{i_\mathrm{s}+1}$, and its binary complement vector $\overline{\tilde{\boldsymbol{v}}_{i_\mathrm{s}+1}}$ [$\tilde{\boldsymbol{v}}_{f_4}$ and $\overline{\tilde{\boldsymbol{v}}_{f_4}}$ in Fig. 7(b)] are prepared. When we define

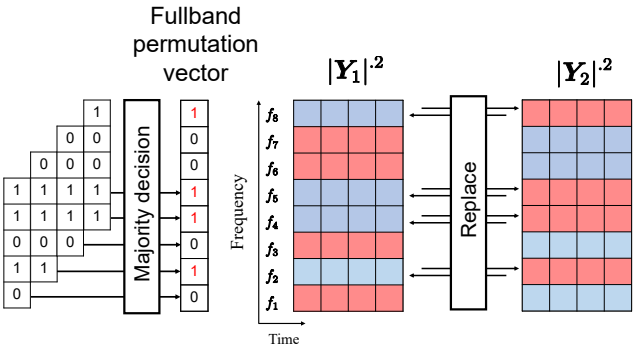

Fig. 8. Solving permutation problem by replacing frequency-wise components on the basis of the fullband permutation vector.

a part of frequency bins in $\boldsymbol{u}$ as

$$\check{\boldsymbol{u}}_i = (u_{i-\Omega}, u_{i-\Omega+1}, \cdots, u_{i+\Omega-1})^{\mathrm{T}} \in \{0,1\}^{2\Omega}, \qquad (22)$$

the two similarities $\mathrm{MSE}(\tilde{\boldsymbol{v}}_{i_s+1}, \check{\boldsymbol{u}}_{i_s+1})$ and $\mathrm{MSE}(\overline{\tilde{\boldsymbol{v}}_{i_s+1}}, \check{\boldsymbol{u}}_{i_s+1})$ are compared, where $\mathrm{MSE}(\cdot,\cdot)$ returns the MSE value between two input vectors. Then, the vector $\tilde{\boldsymbol{v}}_{i_s+1}$ or $\overline{\tilde{\boldsymbol{v}}_{i_s+1}}$ that minimizes MSE is selected and stored in the memory. The fullband permutation vector $\boldsymbol{u}$ is updated by taking a majority decision using the vectors stored in the memory as shown in Fig. 7(b). Finally, by iterating the above-mentioned step, the complete fullband permutation vector $\boldsymbol{u}$ can be obtained as shown in Fig. 7(c).

It is worth mentioning that the iterative majority decision used in the reconstruction process of $\boldsymbol{u}$ effectively reduces the adverse effect of the DNN prediction errors, similarly to the majority decision in Fig. 5.

F. Replacing Components Based on Fullband Permutation Vector

The fullband permutation vector $\boldsymbol{u}$ is equivalent to the estimate of the permutation matrix $\boldsymbol{P}_i^{-1}$. Thus, the frequency-wise source components can be replaced to solve the permutation problem. This process is illustrated in Fig. 8.

IV. Experiments

A. Conditions

To evaluate the performance of the proposed permutation solver, we conducted BSS experiments in which speech sources were separated. We compared three BSS methods, namely, FDICA with the ideal permutation solver (IPS), ILRMA [9], and FDICA with the proposed permutation solver. The IPS utilizes the oracle (completely separated) source signals for solving the permutation problem, which provides upper-limit performance for FDICA-based BSS. The details of FDICA with IPS can be found in Ref. [11].

We used the "nonpara30" dataset obtained from the Japanese versatile speech (JVS) corpus [13] as training data for the proposed permutation solver. We used 190 speech files (95 files for 46 males and 95 files for 48 females) from the dataset, and 10-s-long two speech files were used as the dry

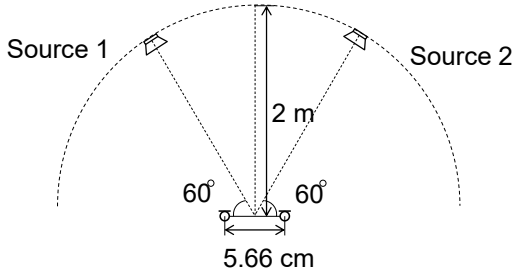

Fig. 9. Recording condition of JR2 impulse response.

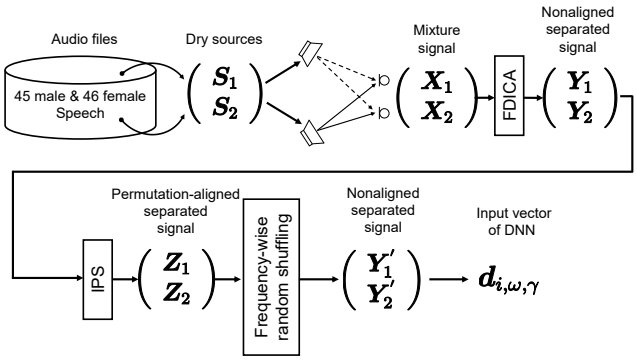

Fig. 10. Process flow of producing input vectors for DNN.

sources. To simulate the multichannel recording, these speech dry sources, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, were convoluted with the JR2 impulse response obtained from the RWCP database [14], where the reverberation time is $T_{60} = 470$ ms. The recording condition of the JR2 impulse response is shown in Fig. 9.

The input vectors for DNN were produced by simulating FDICA separation as depicted in Fig. 10, where the label vectors are obtained from the result of frequency-wise random shuffling. STFT was applied to the observed signals using a 512-ms-long Hamming window with 128-ms-long shifting to produce the mixture spectrograms $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, and the separated signals $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ were estimated by FDICA. The permutation-aligned spectrograms $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ can be obtained by applying IPS to $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$. To simulate the FDICA-based separation with the permutation problem, the frequency components were randomly shuffled, and the simulated non-aligned spectrograms $\boldsymbol{Y}_1'$ and $\boldsymbol{Y}_2'$ were produced. The DNN input vectors and their label vectors were calculated from $\boldsymbol{Y}_1'$ and $\boldsymbol{Y}_2'$, and we prepared 400,000 pairs of input and label vectors as the training data, where the training data was split into 200,000 training and 200,000 validation pairs. The dry sources of test data (four males and four females) were obtained from underdetermined separation tasks (dev1 data) in the SiSEC2011 dataset [15]. These signals were also convoluted with the JR2 impulse response, and 28 observed signals were produced. In DNN training, the batch size was set to 128, and we used the Adam optimizer to train the network. We did not use the dropout technique. Also, we set $\tau = 40$, $\Omega = 15$, and $\eta = 4$, where $\Gamma = 37$. The number

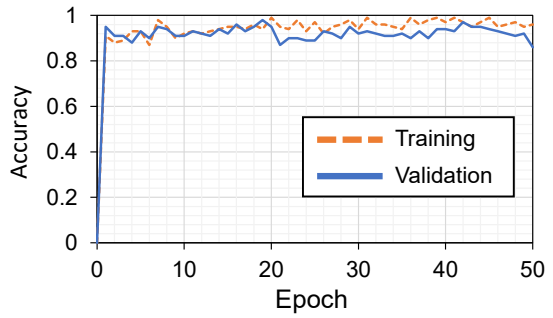

Fig. 11. Accuracy curves of DNN for training and validation datasets.

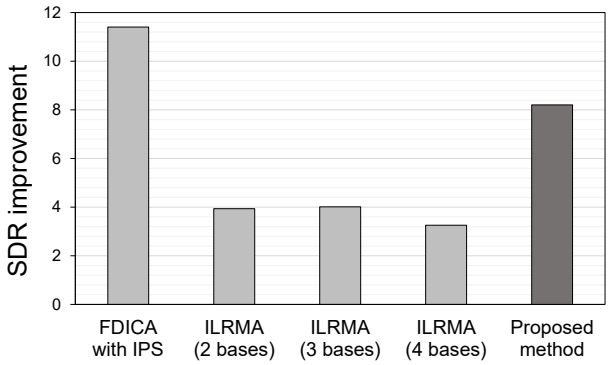

Fig. 12. Average SDR improvements of methods.

of dimensions of the input vector was $40 \times 4 = 160$. We evaluated the improvement of SDR [12], which shows the total separation accuracy including both the degree of separation and the absence of artificial distortion.

B. Results

Fig. 11 shows the accuracy curves of DNN for the training and validation datasets. From the results, we can confirm that the DNN solves the frequency-wise permutation problem with more than 85% accuracy. That is, there is a 15% chance that the DNN fails to estimate the correct permutation. However, the majority decisions along time frames and frequency bins can effectively reduce the adverse effect of these prediction errors.

The SDR improvements of all the methods are shown in Fig. 12, where all the results for 28 observed signals were averaged. As already reported in Ref. [11], the separation performance of ILRMA is less than 4 dB, whereas FDICA with IPS achieves over 10 dB improvement. These results show that ILRMA fails to precisely solve the permutation problem. The proposed method, FDICA with the DNN-based permutation solver, outperforms ILRMA and achieves over 8 dB improvement, which is relatively close to the upper-limit performance of FDICA-based separation. This is because the supervised permutation solver provides better performance for estimating the correct source permutation in each frequency bin.

V. Conclusion

In this paper, we proposed a new DNN-based permutation solver for determined audio source separation using FDICA, where only the two-source mixture case is treated. The DNN model in the proposed permutation solver is trained so that the DNN predicts whether the two input binwise activations of each source are the correct permutation as a binary scalar. In addition, to cope with the prediction errors in the DNN model, the majority decisions along time frames and frequency bins are performed. From the speech separation experiments, FDICA with the proposed permutation solver outperforms the conventional ILRMA. The proposed permutation solver can be applied to only the two-source mixture case. Developing a more general framework of the DNN-based permutation solver for an arbitrary number of sources is our important future work.

References

[1] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.
[2] P. Smaragdis, "Blind separation of convolved mixtures in the Frequency-domain," *Neurocomputing*, vol. 22, pp. 21–34, 1998.
[3] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1–4, pp. 1–24, 2001.
[4] H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee, and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," *IEEE Trans. ASLP*, vol. 14, no. 2, pp. 666–678, 2006.
[5] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of Frequency-domain blind source separation," *IEEE Trans. SAP*, vol. 12, no. 5, pp. 530–538, Sep. 2004.
[6] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Trans. ASLP*, vol. 15, no. 1, pp. 70–79, 2007.
[7] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," *Proc. WASPAA*, pp. 189–192, 2011.
[8] 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.
[9] 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., pp. 125–155. Springer, Cham, 2018.
[10] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. ASLP*, vol. 18, no. 7, pp. 1830–1840, 2010.
[11] D. Kitamura, N. Ono, and H. Saruwatari, "Experimental analysis of optimal window length for independent low-rank matrix analysis," *Proc. EUSIPCO*, pp. 1210–1214, 2017.
[12] 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.
[13] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JVS corpus: free Japanese multi-speaker voice corpus," *arXiv preprint*, 1908.06248, 2019.
[14] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," *Proc. LREC*, pp. 965–968, 2000.
[15] S. Araki, F. Nesta, E. Vincent, Z. Koldovsky, G. Nolte, A. Ziehe and A. Benichoux, "The 2011 signal separation evaluation campaign (SiSEC2011): -Audio source separation," *Proc. LVA/ICA*, pp. 414-422, 2012.