DNN-Based Frequency Component Prediction for Frequency-Domain Audio Source Separation

Rui Watanabe*, Daichi Kitamura*, Hiroshi Saruwatari†, Yu Takahashi‡, and Kazunobu Kondo‡
*National Institute of Technology, Kagawa College, Kagawa 761–8058, Japan
†The University of Tokyo, Tokyo 113–8656, Japan
‡Yamaha Corporation, Shizuoka 430–8650, Japan

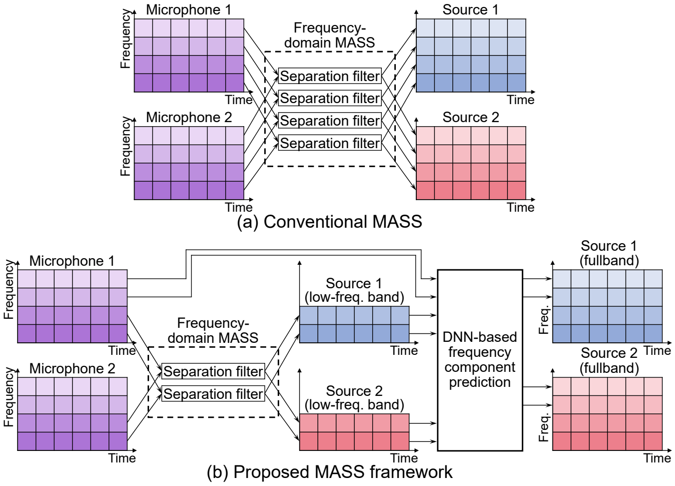
***Abstract*—Multichannel audio source separation (MASS) plays an important role in various audio applications. Frequency-domain MASS algorithms such as multichannel nonnegative matrix factorization achieve better separation quality. However, they require a considerable computational cost for estimating the frequency-wise separation filter. To solve this problem, we propose a new framework combining the MASS algorithms and a simple deep neural network (DNN). In the proposed framework, frequency-domain MASS is performed only in narrowband frequency bins. Then, DNN predicts the separated source components in other frequency bins, where both the observed mixture of all frequency bins and the separated narrowband source components are used as DNN inputs. Our experimental results show the validity of the proposed MASS framework in terms of computational efficiency.**

***Index Terms*—audio source separation, deep neural networks, frequency component prediction**

I. Introduction

Audio source separation (ASS) is a technique to estimate specific audio sources from an observed mixture signal and is a critical preprocessing for *machine listening* (audio understanding by machine), which finds applications in various fields. In particular, multichannel ASS (MASS) has recently attracted attention because typical audio devices (e.g., smart speakers) have a microphone array. One of the popular MASS algorithms is multichannel nonnegative matrix factorization (multichannel NMF: MNMF) [1], [2]. MNMF estimates both a spatial model (acoustic paths from sources to microphones) as a source-frequency-wise spatial covariance matrix (SCM) [3] and a source model as a low-rank time-frequency structure of each source by NMF [4], [5]. Then, the separation filter in each frequency bin is calculated using the estimated spatial and source models, resulting in frequency-domain MASS. Although MNMF separates sources well without any prior information or training, it requires a huge computational cost for estimating the parameters.

Deep neural networks (DNNs) have also been principally used for solving various tasks, and DNN-based single-channel ASS has been widely investigated [6]–[8]. Although DNN-based MASS [9]–[12] has the potential to achieve high-quality separation, it is still challenging to implement them into a consumer product because of their large-scale network architectures.



(a) Conventional MASS

(b) Proposed MASS framework

Fig. 1. Frameworks of (a) conventional and (b) proposed MASS. In proposed framework, frequency-domain MASS is applied to only limited number of frequency bins and other frequency components are predicted by DNN.

In this paper, we propose a new, simple, and efficient MASS framework employing both conventional frequency-domain MASS and DNN-based frequency component prediction. The framework consists of two steps: (a) applying frequency-domain MASS to narrowband (limited number of) frequency bins and (b) predicting the source components in other frequency bins using a small-scale simple DNN to construct fullband estimated sources. These processes are illustrated in Fig. 1. Since the number of frequency bins can be reduced in the MASS step, the total computational cost is reduced if the DNN prediction step is efficient, which can be achieved by using edge-computing devices. In this paper, we only treat MNMF [2] as an example of popular MASS, but any frequency-domain ASS algorithm (e.g., [13]–[15]) is applicable in the proposed framework.

II. Frequency-domain MASS

A. Formulation

Let N and M be the numbers of sources and microphones. The observed multichannel source and mixture signals obtained by the short-time Fourier transform (STFT) are defined as follows:

$$\boldsymbol{s}_{ijn} = (s_{ijn1}, \cdots, s_{ijnm}, \cdots, s_{ijnM})^{\mathrm{T}} \in \mathbb{C}^N, \qquad (1)$$

$$\boldsymbol{x}_{ij} = (x_{ij1}, \cdots, x_{ijm}, \cdots, x_{ijN})^{\mathrm{T}} \in \mathbb{C}^M, \qquad (2)$$

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

where $i=1,2,\ldots,I$, $j=1,2,\ldots,J$, $n=1,2,\ldots,N$, and $m=1,2,\ldots,M$ are the indices of frequency bins, time frames, sources, and microphones, respectively. We also define the notations of the spectrogram matrix for (1) and (2) as $\boldsymbol{S}_{nm}\in\mathbb{C}^{I\times J}$ and $\boldsymbol{X}_m\in\mathbb{C}^{I\times J}$, respectively. The multichannel source signal $\boldsymbol{s}_{ijn}$ is often called *source image*. The observed multichannel mixture signal is assumed to be the sum of source images as $\boldsymbol{x}_{ij}=\sum_n \boldsymbol{s}_{ijn}$.

B. State-of-the-Art MASS: MNMF

In MNMF [2], the following generative model is assumed for the source image:

$$\boldsymbol{s}_{ijn}\sim\mathcal{N}(\boldsymbol{0},\sigma_{ijn}\boldsymbol{H}_{in}), \tag{3}$$

$$\sigma_{ijn}=\sum_k z_{kn}t_{ik}v_{kj}, \tag{4}$$

where $\mathcal{N}(\boldsymbol{0},\sigma_{ijn}\boldsymbol{H}_{in})$ is a zero-mean multivariate Gaussian distribution with the covariance matrix $\sigma_{ijn}\boldsymbol{H}_{in}$. Also, $t_{ik}\geq 0$ and $v_{kj}\geq 0$ are the NMF parameters (basis and activation), $k=1,2,\ldots,K$ is the index of NMF bases, and $z_{kn}\in\mathbb{R}_{[0,1]}$ is a partitioning parameter that clusters K bases into N sources. The parameter σ_{ijn} is called *source model* that corresponds to the expectation of the power spectrogram of each source. The time-invariant matrix $\boldsymbol{H}_{in}\in\mathbb{C}^{M\times M}$ is called *spatial model* or SCM that encodes acoustic paths from the source to all microphones and their spatial spreads.

The parameters σ_{ijn} and $\boldsymbol{H}_{in}$ are optimized on the basis of maximum likelihood estimation. In particular, the update calculation of SCM $\boldsymbol{H}_{in}$ requires a huge computational cost because it includes matrix inversions of size M matrices in each time-frequency slot as follows:

$$\boldsymbol{H}_{in}\leftarrow\boldsymbol{R}_{in}^{-1/2}\left(\boldsymbol{R}_{in}^{1/2}\boldsymbol{A}_{in}\boldsymbol{R}_{in}^{1/2}\right)^{1/2}\boldsymbol{R}_{in}^{-1/2}, \tag{5}$$

$$\boldsymbol{R}_{in}=\sum_{j,k}z_{kn}t_{ik}v_{kj}\boldsymbol{D}_{ij}^{-1}, \tag{6}$$

$$\boldsymbol{D}_{ij}=\sum_n\sigma_{ijn}\boldsymbol{H}_{in}, \tag{7}$$

$$\boldsymbol{A}_{in}=\boldsymbol{H}_{in}\left(\sum_{j,k}z_{kn}t_{ik}v_{kj}\boldsymbol{D}_{ij}^{-1}\boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^{\mathrm{H}}\boldsymbol{D}_{ij}^{-1}\right)\boldsymbol{H}_{in}. \tag{8}$$

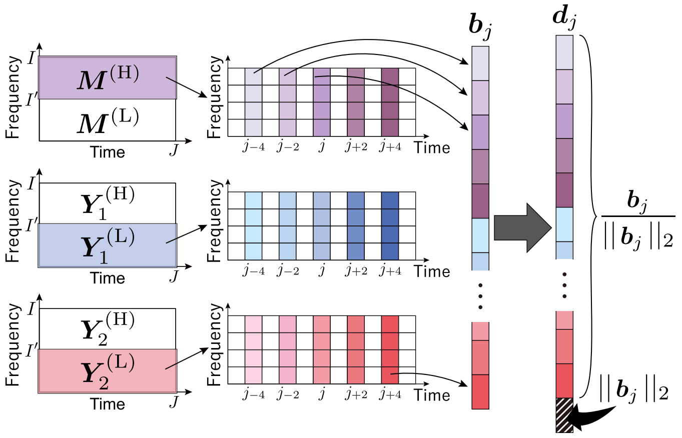
The separated source image $\hat{\boldsymbol{s}}_{ijn}$ is obtained using a multichannel Wiener filter with the estimated parameters σ_{ijn} and $\boldsymbol{H}_{in}$ as

$$\hat{\boldsymbol{s}}_{ijn}=\left(\sigma_{ijn}\boldsymbol{H}_{in}\right)\boldsymbol{D}_{ij}^{-1}\boldsymbol{x}_{ij}. \tag{9}$$

III. Proposed Framework

A. Motivation and Strategy

Although existing frequency-domain MASS methods, including MNMF, can accurately separate the sources, it remains difficult to implement them for a consumer product because of their algorithmic complexity. A large-scale DNN is also not suitable for edge computing on a memory-limited small device such as a smart speaker.



$\boldsymbol{b}_j$
$\boldsymbol{d}_j$
Frequency
I
I'
$\boldsymbol{M}^{(\mathrm{H})}$
$\boldsymbol{M}^{(\mathrm{L})}$
Time
J
$j{-}4$
$j{-}2$
j
$j{+}2$
$j{+}4$
$\boldsymbol{Y}_1^{(\mathrm{H})}$
$\boldsymbol{Y}_1^{(\mathrm{L})}$
$\boldsymbol{Y}_2^{(\mathrm{H})}$
$\boldsymbol{Y}_2^{(\mathrm{L})}$
$\dfrac{\boldsymbol{b}_j}{\|\boldsymbol{b}_j\|_2}$
$\|\boldsymbol{b}_j\|_2$

Fig. 2. Input vector for DNN.

To solve this problem, we propose a new framework combining frequency-domain MASS and frequency component prediction using a small-scale simple DNN. In this framework, as shown in Fig. 1, MASS is applied to only a limited number of frequency bins (low-frequency band in Fig. 1). Then, the separated source components of other frequency bins (high-frequency band in Fig. 1) are predicted using DNN, where the input data of DNN are both (a) the narrowband separated source components and (b) the other frequency components of the mixture signal. Since we utilize not only (a) but also (b) in the DNN-prediction step, the source components can be predicted with high accuracy even if we use a small-scale simple DNN architecture, which may be implementable into a typical consumer product.

For the sake of simplicity, this paper only treats a two-source mixture case ($N=2$). However, the proposed framework can be applied to mixtures of more than two sources by simply extending the DNN model described below. Also, we only consider to divide spectrograms into low- and high-frequency bands, and MASS is applied to only the low-frequency band as shown in Fig. 1. Note that any frequency bins can be selected as the input in the frequency-limited MASS step.

B. DNN Input

Let $\boldsymbol{Y}_1\in\mathbb{R}_{\geq 0}^{I\times J}$ and $\boldsymbol{Y}_2\in\mathbb{R}_{\geq 0}^{I\times J}$ be the amplitude spectrograms of two source signals, and let $\boldsymbol{M}\in\mathbb{R}_{\geq 0}^{I\times J}$ be the amplitude spectrogram of their mixture signal. Since we have the multichannel source and observation signals, the monaural amplitude spectrograms $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, and $\boldsymbol{M}$ are defined using a reference microphone, e.g., $\boldsymbol{Y}_1=\mathrm{abs}(\boldsymbol{S}_{11})$, $\boldsymbol{Y}_2=\mathrm{abs}(\boldsymbol{S}_{21})$, and $\boldsymbol{M}=\mathrm{abs}(\boldsymbol{X}_1)$, where $\mathrm{abs}(\cdot)$ returns a matrix with the element-wise absolute value of the input matrix. In addition, let $i=I'$ be a boundary frequency bin. As shown in Fig. 2, the spectrogram $\boldsymbol{Y}_n$ is divided into the low- and high-frequency bands as $\boldsymbol{Y}_n^{(\mathrm{L})}\in\mathbb{R}_{\geq 0}^{I'\times J}$ and $\boldsymbol{Y}_n^{(\mathrm{H})}\in\mathbb{R}_{\geq 0}^{(I-I')\times J}$, respectively. Similarly, the narrowband spectrograms of $\boldsymbol{M}$ are defined as $\boldsymbol{M}^{(\mathrm{L})}\in\mathbb{R}_{\geq 0}^{I'\times J}$ and $\boldsymbol{M}^{(\mathrm{H})}\in\mathbb{R}_{\geq 0}^{(I-I')\times J}$.

The DNN model predicts the high-frequency narrowband source components, which are not separated in the MASS step, from the low-frequency narrowband separated sources, $\boldsymbol{Y}_1^{(\mathrm{L})}$

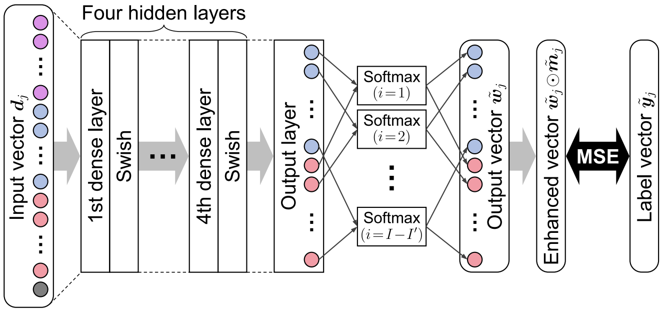

Four hidden layers
Input vector $\boldsymbol{d}_j$
1st dense layer
Swish
4th dense layer
Swish
Output layer
Softmax $(i=1)$
Softmax $(i=2)$
Softmax $(i=I-I')$
Output vector $\tilde{\boldsymbol{w}}_j$
Enhanced vector $\tilde{\boldsymbol{w}}_j \odot \tilde{\boldsymbol{m}}_j$
MSE
Label vector $\tilde{\boldsymbol{y}}_j$

Fig. 3. DNN architecture.

and $\boldsymbol{Y}_2^{(\mathrm{L})}$, and the high-frequency narrowband mixture $\boldsymbol{M}^{(\mathrm{H})}$. More precisely, the DNN outputs two soft masks that enhance $\boldsymbol{Y}_1^{(\mathrm{H})}$ and $\boldsymbol{Y}_2^{(\mathrm{H})}$ from $\boldsymbol{M}^{(\mathrm{H})}$.

The input vector of the DNN model is shown in Fig. 2. The high-frequency mixture and the low-frequency source vectors at a time frame j are defined as

$$\boldsymbol{m}_j^{(\mathrm{H})} = \left(m_{1j}^{(\mathrm{H})}, m_{2j}^{(\mathrm{H})}, \cdots, m_{(I-I')j}^{(\mathrm{H})}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{I-I'}, \tag{10}$$

$$\boldsymbol{y}_{nj}^{(\mathrm{L})} = \left(y_{n1j}^{(\mathrm{L})}, y_{n2j}^{(\mathrm{L})}, \cdots, y_{nI'j}^{(\mathrm{L})}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{I'}, \tag{11}$$

where $m_{ij}^{(\mathrm{H})}$ and $y_{nij}^{(\mathrm{L})}$ are the ij elements of $\boldsymbol{M}^{(\mathrm{H})}$ and $\boldsymbol{Y}_n^{(\mathrm{L})}$, respectively. When the DNN predicts the high-frequency source components at time frame j, the components around j are also important. Thus, we concatenate the neighboring time frames[1] of (10) and (11) to define the input vector for DNN as follows:

$$\overline{\boldsymbol{m}}_j^{(\mathrm{H})} = \left(\boldsymbol{m}_{j-2c}^{(\mathrm{H})}{}^{\mathrm{T}}, \cdots, \boldsymbol{m}_j^{(\mathrm{H})}{}^{\mathrm{T}}, \cdots, \boldsymbol{m}_{j+2c}^{(\mathrm{H})}{}^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{12}$$

$$\overline{\boldsymbol{y}}_{nj}^{(\mathrm{L})} = \left(\boldsymbol{y}_{n(j-2c)}^{(\mathrm{L})}{}^{\mathrm{T}}, \cdots, \boldsymbol{y}_{nj}^{(\mathrm{L})}{}^{\mathrm{T}}, \cdots, \boldsymbol{y}_{n(j+2c)}^{(\mathrm{L})}{}^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{13}$$

$$\boldsymbol{b}_j = \left(\overline{\boldsymbol{m}}_j^{(\mathrm{H})}{}^{\mathrm{T}}, \overline{\boldsymbol{y}}_{1j}^{(\mathrm{L})}{}^{\mathrm{T}}, \overline{\boldsymbol{y}}_{2j}^{(\mathrm{L})}{}^{\mathrm{T}}\right) \in \mathbb{R}_{\geq 0}^{(2C+1)(I+I')}, \tag{14}$$

where $c = 0, 1, \ldots, C$ is the index of neighboring time frames. The vector $\boldsymbol{b}_j$ is normalized to make the DNN training stable, where the normalization coefficient is provided so that the volume can still be exploited as

$$\boldsymbol{d}_j = \left(\frac{1}{\|\boldsymbol{b}_j\|_2}\boldsymbol{b}_j^{\mathrm{T}}, \|\boldsymbol{b}_j\|_2\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{(2C+1)(I+I')+1}, \tag{15}$$

where $\|\cdot\|_2$ denotes the L_2 norm. The input vector for the proposed DNN model is (15).

C. DNN Output and Model Training

Fig. 3 shows the architecture of DNN. The network is fully connected and has four hidden layers, where all the hidden layers have the same number of units (dimensions) as in the output layer. Swish [16] is used as an activation function of all the hidden layers. For the output layer, the frequency-wise softmax function is applied because the sum of the soft masks of all the sources must be one in each frequency bin.

Let $\boldsymbol{W}_n \in \mathbb{R}_{[0,1]}^{(I-I')\times J}$ be the soft mask that enhances the high-frequency narrowband nth source components as

$$\boldsymbol{Y}_n^{(\mathrm{H})} \approx \boldsymbol{W}_n \odot \boldsymbol{M}^{(\mathrm{H})}, \tag{16}$$

where $\odot$ denotes the element-wise product. The output vector of DNN, $\tilde{\boldsymbol{w}}_j$, is a concatenation of column vectors of the soft mask matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ at the time frame j as follows:

$$\tilde{\boldsymbol{w}}_j = \left(\boldsymbol{w}_{1j}^{\mathrm{T}}, \boldsymbol{w}_{2j}^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}_{[0,1]}^{2(I-I')}, \tag{17}$$

$$\boldsymbol{w}_{nj} = \left(w_{n1j}, w_{n2j}, \cdots, w_{n(I-I')j}\right)^{\mathrm{T}} \in \mathbb{R}_{[0,1]}^{I-I'}, \tag{18}$$

where w_{nij} is the element of $\boldsymbol{W}_n$ and $\sum_n w_{nij} = 1$ is ensured for all i and j by the softmax function of the output layer. The label (ground truth) vector of the high-frequency source components is obtained as

$$\tilde{\boldsymbol{y}}_j = \left(\boldsymbol{y}_{1j}^{(\mathrm{H})}{}^{\mathrm{T}}, \boldsymbol{y}_{2j}^{(\mathrm{H})}{}^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{2(I-I')}, \tag{19}$$

$$\boldsymbol{y}_{nj}^{(\mathrm{H})} = \left(y_{n1j}^{(\mathrm{H})}, y_{n2j}^{(\mathrm{H})}, \cdots, y_{n(I-I')j}^{(\mathrm{H})}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{I-I'}, \tag{20}$$

where $y_{nij}^{(\mathrm{H})}$ is the element of $\boldsymbol{Y}_n^{(\mathrm{H})}$. Thus, the DNN model is trained so that the following mean squared error (MSE) is minimized:

$$\mathrm{MSE}(\tilde{\boldsymbol{y}}_j, \tilde{\boldsymbol{w}}_j \odot \tilde{\boldsymbol{m}}_j) = \frac{1}{2(I-I')}\|\tilde{\boldsymbol{y}}_j - \tilde{\boldsymbol{w}}_j \odot \tilde{\boldsymbol{m}}_j\|_2^2, \tag{21}$$

$$\tilde{\boldsymbol{m}}_j = \left(\boldsymbol{m}_j^{(\mathrm{H})}{}^{\mathrm{T}}, \boldsymbol{m}_j^{(\mathrm{H})}{}^{\mathrm{T}}\right) \in \mathbb{R}_{\geq 0}^{2(I-I')}. \tag{22}$$

D. Reconstruction of Separated Time-Domain Signals

In the test stage, the low-frequency narrowband separated sources, $\boldsymbol{Y}_1^{(\mathrm{L})}$ and $\boldsymbol{Y}_2^{(\mathrm{L})}$, and the high-frequency narrowband mixture $\boldsymbol{M}^{(\mathrm{H})}$ are input to the pretrained DNN model, and the framewise soft mask $\tilde{\boldsymbol{w}}_j$ is predicted. The estimated high-frequency components are obtained by (16). Thus, the estimated amplitude spectrogram $\boldsymbol{Y}_n$ can be obtained by concatenating $\boldsymbol{Y}_n^{(\mathrm{L})}$ and $\boldsymbol{Y}_n^{(\mathrm{H})}$.

To reconstruct the time-domain signal of $\boldsymbol{Y}_n$, phase information of $\boldsymbol{Y}_n^{(\mathrm{H})}$ must be recovered. In the proposed method, we simply copy the phase spectrogram of the mixture $\boldsymbol{M}^{(\mathrm{H})}$ to both $\boldsymbol{Y}_1^{(\mathrm{H})}$ and $\boldsymbol{Y}_2^{(\mathrm{H})}$, and the inverse STFT is applied to each complex-valued spectrogram.

IV. Experiments

To confirm the validity of the proposed framework, we conducted two experiments: (a) comparison of net prediction ability of frequency components based on DNNs and (b) MASS experiment using a music mixture of drums (Dr.) and vocals (Vo.).

A. Comparison of Net Prediction Ability

1) Conditions: In this experiment, the validity of the proposed DNN model was confirmed by comparing net prediction ability of frequency components based on DNNs, namely, we evaluated how much the prediction was improved by adding the narrowband mixture $\boldsymbol{M}^{(\mathrm{H})}$ to the DNN input. We trained two DNN models: *DNN w/o mixture* and *DNN w/ mixture*.

[1] The reason we skip a time frame as $j-2, j, j+2$ is that adjacent time frames include redundant information by half shifting in the STFT.

DNN w/o mixture can be interpreted as a simple bandwidth expansion based on DNN [17], [18] because this model predicts the high-frequency narrowband source $\boldsymbol{P}^{(\mathrm{H})} \in \mathbb{R}_{\geq 0}^{(I-I')\times J}$ from only the low-frequency narrowband source $\boldsymbol{P}^{(\mathrm{L})} \in \mathbb{R}_{\geq 0}^{I'\times J}$, where $\boldsymbol{P}^{(\mathrm{H})}$ and $\boldsymbol{P}^{(\mathrm{L})}$ are the amplitude spectrograms of a source signal. The input and label vectors of DNN w/o mixture, $\boldsymbol{q}_j$ and $\boldsymbol{p}^{(\mathrm{H})}$, are respectively defined as

$$\boldsymbol{q}_j = \left(\frac{1}{\|\overline{\boldsymbol{p}}_j\|_2}\overline{\boldsymbol{p}}_j^{\mathrm{T}}, \|\overline{\boldsymbol{p}}_j\|_2\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{(2C+1)I'+1}, \tag{23}$$

$$\overline{\boldsymbol{p}}_j^{(\mathrm{L})} = \left(\boldsymbol{p}_{(j-2c)}^{(\mathrm{L})}{}^{\mathrm{T}}, \cdots, \boldsymbol{p}_j^{(\mathrm{L})\mathrm{T}}, \cdots, \boldsymbol{p}_{(j+2c)}^{(\mathrm{L})}{}^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{24}$$

$$\boldsymbol{p}_j^{(\mathrm{L})} = \left(p_{1j}^{(\mathrm{L})}, p_{2j}^{(\mathrm{L})}, \cdots, p_{I'j}^{(\mathrm{L})}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{I'}, \tag{25}$$

$$\boldsymbol{p}_j^{(\mathrm{H})} = \left(p_{1j}^{(\mathrm{H})}, p_{2j}^{(\mathrm{H})}, \cdots, p_{(I-I')j}^{(\mathrm{H})}\right)^{\mathrm{T}} \in \mathbb{R}_{\geq 0}^{I-I'}, \tag{26}$$

where $p_{ij}^{(\mathrm{L})}$ and $p_{ij}^{(\mathrm{H})}$ are the elements of $\boldsymbol{P}^{(\mathrm{L})}$ and $\boldsymbol{P}^{(\mathrm{H})}$, respectively. Phase information of $\boldsymbol{P}^{(\mathrm{H})}$ was recovered by applying the GriffinLim algorithm [19]. Since DNN w/o mixture does not require multiple sources as the input, the training was carried out using one source, resulting in a source-dependent bandwidth expansion model. Thus, two DNNs w/o mixture for Dr. and Vo. were prepared. DNN w/ mixture is the proposed DNN model explained in Sect. III, which requires $\boldsymbol{Y}_1^{(\mathrm{L})}$, $\boldsymbol{Y}_2^{(\mathrm{L})}$, and $\boldsymbol{M}^{(\mathrm{H})}$ as the input and predicts the soft masks for obtaining $\boldsymbol{Y}_1^{(\mathrm{H})}$ and $\boldsymbol{Y}_2^{(\mathrm{H})}$. In this experiment, $\boldsymbol{Y}_1^{(\mathrm{L})}$ and $\boldsymbol{Y}_2^{(\mathrm{L})}$ in DNN w/ mixture were set to completely separated source components. This condition simulates the situation that the MASS step in the proposed framework ideally separates the sources in each frequency bin.

The DNN models were trained with Dr. and Vo. of 100 songs obtained from the SiSEC2016 [20] MUS dataset. The hamming window length and its shift length in STFT were set to 128 ms and 64 ms, respectively, where $I = 1025$ (8 kHz). The input and label vectors ($\boldsymbol{q}_j$ and $\boldsymbol{p}_j^{(\mathrm{H})}$ for DNN w/o mixture and $\boldsymbol{d}_j$ and $\tilde{\boldsymbol{y}}_j$ for DNN w/ mixture) were produced as the training pairs, where the mixture $\boldsymbol{M}$ is the sum of the two dry sources in the time domain. The boundary frequency bin was set to $I' = 512$ (4 kHz), and the four neighboring time frames were input ($C = 2$). We used Adam [21] to optimize the DNN model, and its hyperparameters were $\varepsilon = 1.0 \times 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\eta = 0.001$. The minibatch size was 128, and the number of epochs was 1000.

As the test sources of Dr. and Vo., we used four songs shown in Table I from the SiSEC2011 [23] dataset. We evaluated sources-to-artifacts ratio (SAR) [24] of the reconstructed fullband source signals, where SAR shows the absence of artificial distortion.

2) Results: Table II shows SARs of the fullband source signals predicted by each DNN model. Although DNN w/o mixture is a source-dependent bandwidth expansion model, the proposed DNN model outperforms DNN w/o mixture owing to the utilization of the high-frequency narrowband mixture. This fact shows the validity of the proposed DNN model. The efficiency of the proposed MASS framework combining MNMF and DNN w/ mixture is evaluated in the next subsection.

TABLE I
SONG NAMES OF DRY SOURCE IN TEST DATASET

Song ID	Song name	Signal length [s]
1	dev1__bearlin-roads	14.0
2	dev2__another_dreamer-the_ones_we_love	25.0
3	dev2__fort_minor-remember_the_name	24.0
4	dev2__ultimate_nz_tour	18.0

TABLE II
SARS OF PREDICTED FULLBAND SOURCE SIGNAL

Song ID	DNN w/o mixture	DNN w/ mixture
1	Dr.: 21.1 dB	Dr.: **28.0** dB
	Vo.: 21.8 dB	Vo.: **31.5** dB
2	Dr.: **22.0** dB	Dr.: 21.8 dB
	Vo.: 12.7 dB	Vo.: **19.6** dB
3	Dr.: 15.0 dB	Dr.: **20.4** dB
	Vo.: 11.2 dB	Vo.: **18.5** dB
4	Dr.: 11.0 dB	Dr.: **18.2** dB
	Vo.: 10.4 dB	Vo.: **15.3** dB

B. MASS Experiment Using Music Mixture

1) Conditions: In this experiment, we produced a mixture of Dr. and Vo. observed by two microphones ($N = M = 2$), where RWCP [22] E2A impulse responses shown in Fig. 4 were convoluted to each dry source. As the dry sources of Dr. and Vo., we used the same sources as shown in Table I. In the MASS step, we used MNMF with 13 NMF bases ($K = 13$), which gave better separation results in this experiment. The initial SCMs were set to identity matrices, and all the other parameters were initialized with random values. As the measure of source separation performance, we used source-to-distortion ratio (SDR) [24], which shows both the degree of separation and the quality of the separated source. The calculations in MNMF were carried out on an AMD Ryzen 7 2700X CPU. The DNN prediction was calculated on an NVIDIA GeForce RTX2080Ti GPU.

2) Results: Fig. 5 shows the averaged SDRs of MNMF with a fullband mixture (*fullband MNMF*) and MNMF with a narrowband mixture followed by the DNN prediction (*proposed method*). The horizontal axis indicates the average elapsed time of each method, where the results of the proposed method include the calculation time for the DNN prediction, which is less than 0.1 s. Since the elapsed time depends on the number of iterations of the MNMF parameter update (denoted as L), we performed the proposed method using various numbers of iterations in MNMF, that is, $L = 10, 20, \ldots$. Thus, each plot of the proposed method in Fig. 5 shows the result for each of the L conditions. Fullband MNMF was also performed more than $L = 300$ iterations, and the SDR behaviors are also shown in Fig. 5. In addition, MNMF was performed 10 times with different pseudorandom seeds in both fullband MNMF and the proposed method, and the average SDRs are shown in Fig. 5.

From these results, we can confirm that the proposed method can reduce the computational cost by almost half compared with fullband MNMF. For example, in Fig. 5 (d),

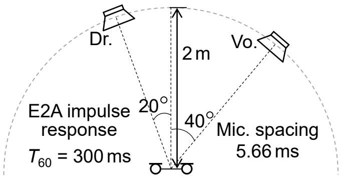


Fig. 4. Impulse responses used in experiment.

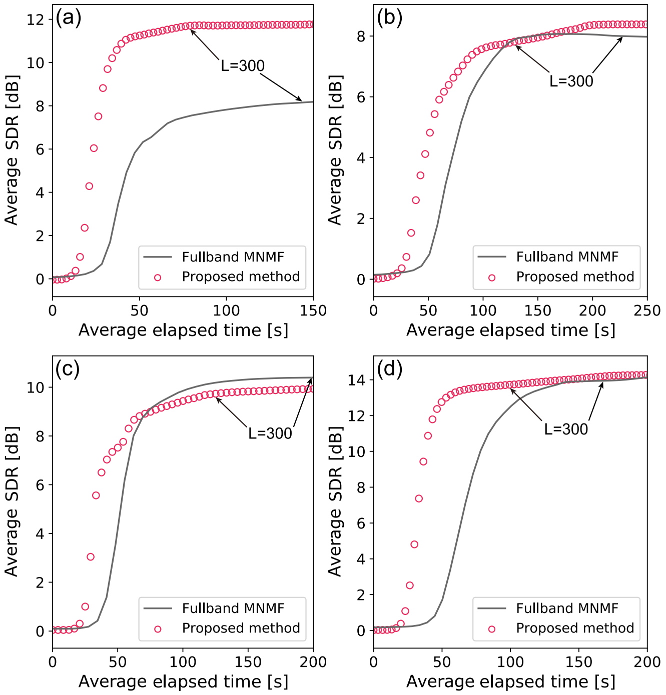


Fig. 5. Average SDRs and their average elapse times: (a) song ID=1, (b) song ID=2, (c) song ID=3, and (d) song ID=4. Each plot of proposed method shows result for each of L conditions (numbers of iterations in narrowband MNMF).

the proposed method achieves 13 dB in less than 50 s, whereas fullband MNMF converged to 13 dB in 120 s. This is because the number of frequency bins in MNMF is reduced from $I = 1025$ to $I - I' = 512$. Also, at the converged point, the proposed method outperforms fullband MNMF in Fig. 5 (a), (b) and (d). This performance of the proposed method was obtained owing to the accurate estimation of the soft masks based on the training with 100 songs.

V. Conclusion

In this paper, we presented a new computationally efficient source separation framework combining an existing ASS technique and a simple-DNN-based frequency component prediction framework. The proposed framework can reduce the computational cost by decimating frequency bins for audio source separation and predicting the source components in the decimated frequency bins. From the experimental results, the validity of the proposed framework was confirmed by achieving faster ASS than by using the conventional fullband ASS algorithm. The proposed framework can easily be extended to various situations, e.g., ASS is applied once in every few frequency bins or is applied to only the valid frequency bins that are effective for source separation.

References

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

[2] 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.

[3] 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.

[4] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[5] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[6] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," *Proc. ICASSP*, pp. 3734–3738, 2014.

[7] P.-S. Huang, M. Kim, M. H.-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. ASLP*, vol. 23, no. 12, pp. 2136–2147, 2015.

[8] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: discriminative embeddings for segmentation and separation," *Proc. ICASSP*, pp. 31–35, 2016.

[9] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," *Proc. ICASSP*, pp. 116–120, 2015.

[10] 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.

[11] T. Nakatani, N. Ito, T. Higuchi, S. Araki, and K. Kinoshita, "Integrating DNN-based and spatial clustering-based mask estimation for robust MVDR beamforming," *Proc. ICASSP*, pp. 286–290, 2017.

[12] N. Makishima, S. Mogami, N. Takamune, D. Kitamura, H. Sumino, S. Takamichi, H. Saruwatari, and N. Ono, "Independent deeply learned matrix analysis for determined audio source separation," *IEEE/ACM Trans. ASLP*, vol. 27, no. 10, pp. 1601–1615, 2019.

[13] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Trans. ASLP*, vol. 19, no. 3, pp. 516–527, 2010.

[14] 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.

[15] K. Yatabe and D. Kitamura, "Time-frequency-masking-based determined BSS with application to sparse IVA," *Proc. ICASSP*, pp. 715–719, 2019.

[16] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint*, arXiv:1710.05941, 2017.

[17] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," *Proc. ICASSP*, pp. 4395–4399, 2015.

[18] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, "Speech bandwidth expansion based on deep neural networks," *Proc. INTERSPEECH*, pp. 2593–2597, 2015.

[19] D. Griffin and J. Lim, "Signal estimation from modified shorttime Fourier transform," *IEEE Trans. ASSP*, vol. 32, no. 2, pp. 236–243, 1984.

[20] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," *Proc. LVA/ICA*, pp. 323–332, 2012.

[21] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint*, arXiv:1412.6980, 2014.

[22] 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.

[23] S. Araki, F. Nesta, E. Vincent, Z. Koldovský, G. Nolte, A. Ziehe and A. Benichoux, "The 2011 signal separation evaluation campaign (SiSEC2011):-audio source separation," *Proc. Latent Variable Analysis and Signal Separation*, pp. 414–422, 2012.

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