LETTER **Communicated by Alexey Ozerov**

Supervised Determined Source Separation with Multichannel Variational Autoencoder

Hirokazu Kameoka
hirokazu.kameoka.uh@hco.ntt.co.jp
Nippon Telegraph and Telephone Corporation, Kanagawa, 243-0198, Japan

Li Li
lili@mmlab.cs.tsukuba.ac.jp
Shota Inoue
s1920622@s.tsukuba.ac.jp
Shoji Makino
maki@tara.tsukuba.ac.jp
University of Tsukuba, Ibaraki, 305-8577, Japan

This letter proposes a multichannel source separation technique, the multichannel variational autoencoder (MVAE) method, which uses a conditional VAE (CVAE) to model and estimate the power spectrograms of the sources in a mixture. By training the CVAE using the spectrograms of training examples with source-class labels, we can use the trained decoder distribution as a universal generative model capable of generating spectrograms conditioned on a specified class index. By treating the latent space variables and the class index as the unknown parameters of this generative model, we can develop a convergence-guaranteed algorithm for supervised determined source separation that consists of iteratively estimating the power spectrograms of the underlying sources, as well as the separation matrices. In experimental evaluations, our MVAE produced better separation performance than a baseline method.

1 Introduction

Blind source separation (BSS) is a technique for separating out individual source signals from microphone array inputs when the transfer characteristics between the sources and microphones are unknown. The frequency-domain BSS approach provides the flexibility of allowing us to utilize various models for the time-frequency representations of source signals and array responses. For example, independent vector analysis (IVA) (Kim, Eltoft, & Lee, 2006; Hiroe, 2006) allows us to efficiently solve frequency-wise source separation and permutation alignment in a joint manner by assuming that the magnitudes of the frequency components originating from the same source tend to vary coherently over time.

Neural Computation **31, 1891–1914** (2019)
doi:10.1162/neco_a_01217

With a different approach, multichannel extensions of nonnegative matrix factorization (NMF) have attracted a lot of attention in recent years (Ozerov & Févotte, 2010; Kameoka, Yoshioka, Hamamura, Le Roux, & Kashino, 2010; Sawada, Kameoka, Araki, & Ueda, 2013; Kitamura, Ono, Sawada, Kameoka, & Saruwatari, 2016, 2017). NMF was originally applied to music transcription and monaural source separation tasks (Smaragdis, 2003; Févotte, Bertin, & Durrieu, 2009). The idea is to approximate the power (or magnitude) spectrogram of a mixture signal, interpreted as a nonnegative matrix, as a product of two nonnegative matrices. This amounts to assuming that the power spectrum of a mixture signal observed at each time frame can be approximated by a linear sum of a limited number of basis spectra scaled by time-varying amplitudes. Multichannel NMF (MNMF) is an extension of this approach to a multichannel case to allow the use of spatial information as an additional clue to separation. It can also be viewed as an extension of frequency-domain BSS that allows the use of spectral templates as a clue for jointly solving frequency-wise source separation and permutation alignment.

The original MNMF (Ozerov & Févotte, 2010) was formulated under a general problem setting where sources can outnumber microphones and a determined version of MNMF was subsequently proposed (Kameoka et al., 2010). While the determined version is applicable only to determined cases, it allows the implementation of a significantly faster algorithm than the general version. The determined MNMF framework was later called "independent low-rank matrix analysis (ILRMA)" (Kitamura et al., 2017). Kitamura et al. (2016) discussed the theoretical relation of MNMF to IVA, which has naturally allowed for the incorporation of the fast update rule of the separation matrix developed for IVA, called "iterative projection (IP)" (Ono, 2011), into the parameter optimization process in ILRMA. It has been shown that this has contributed not only to accelerating the entire optimization process but also to improving the separation performance. One important feature of ILRMA is that the log likelihood to be maximized is guaranteed to be nondecreasing at each iteration of the algorithm. However, one drawback is that it can fail to work for sources with spectrograms that do not comply with the NMF model.

As an alternative to the NMF model, some attempts have recently been made to use deep neural networks (DNNs) for modeling the spectrograms of sources for multichannel source separation (Nugraha, Liutkus, & Vincent, 2016; Mogami et al., 2018). The idea is to replace the process for estimating the power spectra of source signals in a source separation algorithm with the forward computations of pretrained DNNs. This can be viewed as a process of refining the estimates of the power spectra of the source signals at each iteration of the algorithm. While this approach is particularly appealing in that it can take advantage of the strong representation power of DNNs for estimating the power spectra of source signals, one weakness

is that unlike ILRMA, the log likelihood is not guaranteed to be nondecreasing at each iteration of the algorithm.

To address the drawbacks of the methods mentioned above, we propose a multichannel source separation method using variational autoencoders (VAEs) (Kingma & Welling, 2014; Kingma, Rezendey, Mohamedy, & Welling, 2014) for source spectrogram modeling. It should be noted that a preprint paper on this work has already been made publicly available (Kameoka, Li, Inoue, & Makino, 2018). While there have recently been some attempts to apply VAEs to monaural and multichannel speech enhancement (Bando, Mimura, Itoyama, Yoshii, & Kawahara, 2018; Leglaive, Girin, & Horaud, 2018, 2019; Sekiguchi, Bando, Yoshii, & Kawahara, 2018), to the best of our knowledge, our work is the first to propose the application of VAEs to multichannel source separation.

2 Problem Formulation

We consider a situation where J source signals are captured by I microphones. Let $x_i(f,n)$ and $s_j(f,n)$ be 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. We denote the vectors containing $x_1(f,n),\ldots,x_I(f,n)$ and $s_1(f,n),\ldots,s_J(f,n)$ by

$$\mathbf{x}(f,n) = [x_1(f,n),\ldots,x_I(f,n)]^{\mathsf{T}} \in \mathbb{C}^I, \tag{2.1}$$

$$\mathbf{s}(f,n) = [s_1(f,n),\ldots,s_J(f,n)]^{\mathsf{T}} \in \mathbb{C}^J, \tag{2.2}$$

where $(\cdot)^{\mathsf{T}}$ denotes transpose. When the length of the acoustic impulse response from a source to a microphone is sufficiently shorter than the frame length of the STFT, $\mathbf{x}(f,n)$ can be approximated fairly well by an instantaneous mixture in the frequency domain,

$$\mathbf{x}(f,n) = \mathbf{A}(f)\mathbf{s}(f,n), \tag{2.3}$$

where $\mathbf{A}(f)$ is called a mixing matrix. In a particular case where $I = J$ and $\mathbf{A}(f)$ is invertible, we can use a separation system of the form

$$\mathbf{s}(f,n) = \mathbf{W}^{\mathsf{H}}(f)\mathbf{x}(f,n), \tag{2.4}$$

$$\mathbf{W}(f) = [\mathbf{w}_1(f),\ldots,\mathbf{w}_I(f)], \tag{2.5}$$

to describe the relationship between $\mathbf{x}(f,n)$ and $\mathbf{s}(f,n)$ where $\mathbf{W}^{\mathsf{H}}(f) = \mathbf{A}^{-1}(f)$ is called the separation matrix. $(\cdot)^{\mathsf{H}}$ denotes Hermitian transpose. The aim of BSS methods is to estimate $\mathbf{W}^{\mathsf{H}}(f)$ solely from the observations $\mathbf{x}(f,n)$.

Let us now assume that $s_j(f,n)$ independently follows a zero-mean complex gaussian distribution with power spectral density $v_j(f,n) = \mathbb{E}[|s_j(f,n)|^2]$:

$$s_j(f,n) \sim \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, v_j(f,n)). \tag{2.6}$$

Equation 2.6 is usually called the local gaussian model (LGM) (Févotte & Cardoso, 2005; Vincent, Arberet, & Gribonval, 2009). When $s_j(f,n)$ and $s_{j'}(f,n)$ $(j \neq j')$ are independent, $\mathbf{s}(f,n)$ follows

$$\mathbf{s}(f,n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{s}(f,n)|\mathbf{0}, \mathbf{V}(f,n)), \tag{2.7}$$

where $\mathbf{V}(f,n)$ is a diagonal matrix with diagonal entries $v_1(f,n), \ldots, v_I(f,n)$. From 2.4 and 2.6, we can show that $\mathbf{x}(f,n)$ follows

$$\mathbf{x}(f,n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{x}(f,n)|\mathbf{0}, (\mathbf{W}^{\mathsf{H}}(f))^{-1}\mathbf{V}(f,n)\mathbf{W}(f)^{-1}). \tag{2.8}$$

Hence, the log likelihood of the separation matrices $\mathcal{W} = \{\mathbf{W}(f)\}_f$ given the observed mixture signals $\mathcal{X} = \{\mathbf{x}(f,n)\}_{f,n}$ is given by

$$\begin{aligned}\log p(\mathcal{X}|\mathcal{W}, \mathcal{V}) &\stackrel{c}{=} 2N \sum_f \log|\det \mathbf{W}^{\mathsf{H}}(f)| \\ &\quad - \sum_{f,n}\sum_j \left(\log v_j(f,n) + \frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{v_j(f,n)} \right),\end{aligned} \tag{2.9}$$

where $\stackrel{c}{=}$ denotes equality up to constant terms. If we individually treat $v_j(f,n)$ as a free parameter, all the variables in equation 2.9 will be indexed by frequency f. The optimization problem will thus be split into frequency-wise source separation problems. Under this problem setting, the permutation of the separated components in each frequency cannot be uniquely determined. Thus, we usually need to group together the separated components of different frequency bins that originate from the same source after we obtain $\mathcal{W}$. This process is called permutation alignment. However, it is preferable to solve permutation alignment and source separation jointly since the clues used for permutation alignment can also be helpful for source separation. If there is a certain assumption, constraint, or structure that we can incorporate into $v_j(f,n)$, it can help eliminate the permutation ambiguity during the estimation of $\mathcal{W}$. One such example is the NMF model, which expresses $v_j(f,n)$ as the linear sum of spectral templates: $b_{j,1}(f), \ldots, b_{j,K_j}(f) \geq 0$ scaled by time-varying magnitudes

$h_{j,1}(n), \ldots, h_{j,K_j}(n) \geq 0$:

$$v_j(f,n) = \sum_{k=1}^{K_j} b_{j,k}(f) h_{j,k}(n). \tag{2.10}$$

ILRMA is a BSS framework that incorporates this model into the log likelihood, equation 2.9 (Kameoka et al., 2010; Kitamura et al., 2016, 2017). Here, we consider a particular case where $K_j = 1$ and $b_{j,k}(f) = 1$ for all j in equation 2.10, which means each source has only one flat-shaped spectral template. Under this constraint, we can show that both assuming $s_j(0,n), \ldots, s_j(F,n)$ independently follow equation 2.6 and assuming the norm $r_j(n) = \sqrt{\sum_f |s_j(f,n)|^2}$ follows a complex gaussian distribution with time-varying variance $h_j(n)$ result in the same log likelihood (Ozerov & Kameoka, 2018). This is analogous to the assumption employed by IVA where the norm $r_j(n)$ is assumed to follow a supergaussian distribution. Kitamura et al. (2016) showed that ILRMA can significantly outperform IVA in terms of source separation ability. This fact implies that within the LGM-based BSS framework, the stronger the representation power of a power spectrogram model becomes, the better the source separation performance we can expect to obtain.

3 Related Work

3.1 ILRMA. The optimization algorithm of ILRMA consists of iteratively updating $\mathcal{W}$, $\mathcal{B} = \{b_{j,k}(f)\}_{j,k,f}$ and $\mathcal{H} = \{h_{j,k}(n)\}_{j,k,n}$ so that equation 2.9 is nondecreasing at each iteration (Kameoka et al., 2010; Kitamura et al., 2016, 2017). To update $\mathcal{W}$, we can use the natural gradient method or IP. The IP-based update rule for $\mathcal{W}$ (Ono, 2011) is given as

$$\mathbf{w}_j(f) \leftarrow (\mathbf{W}^{\mathsf{H}}(f)\mathbf{\Sigma}_j(f))^{-1}\mathbf{e}_j, \tag{3.1}$$

$$\mathbf{w}_j(f) \leftarrow \frac{\mathbf{w}_j(f)}{\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{\Sigma}_j(f)\mathbf{w}_j(f)}, \tag{3.2}$$

where $\mathbf{\Sigma}_j(f) = \frac{1}{N}\sum_n \mathbf{x}(f,n)\mathbf{x}^{\mathsf{H}}(f,n)/v_j(f,n)$ and $\mathbf{e}_j$ denotes the jth column of the $I \times I$ identity matrix. To update $\mathcal{B}$ and $\mathcal{H}$, we can employ the expectation-maximization (EM) algorithm or the majorization-minimization (MM) algorithm. The MM-based update rules for $\mathcal{B}$ and $\mathcal{H}$ can be derived (Kameoka, Goto, & Sagayama, 2006; Nakano, Kameoka, Le Roux, Ono, & Sagayama, 2010; Févotte & Idier, 2011) as

$$b_{j,k}(f) \leftarrow b_{j,k}(f)\sqrt{\frac{\sum_n |y_j(f,n)|^2 h_{j,k}(n)/v_j^2(f,n)}{\sum_n h_{j,k}(n)/v_j(f,n)}}, \tag{3.3}$$

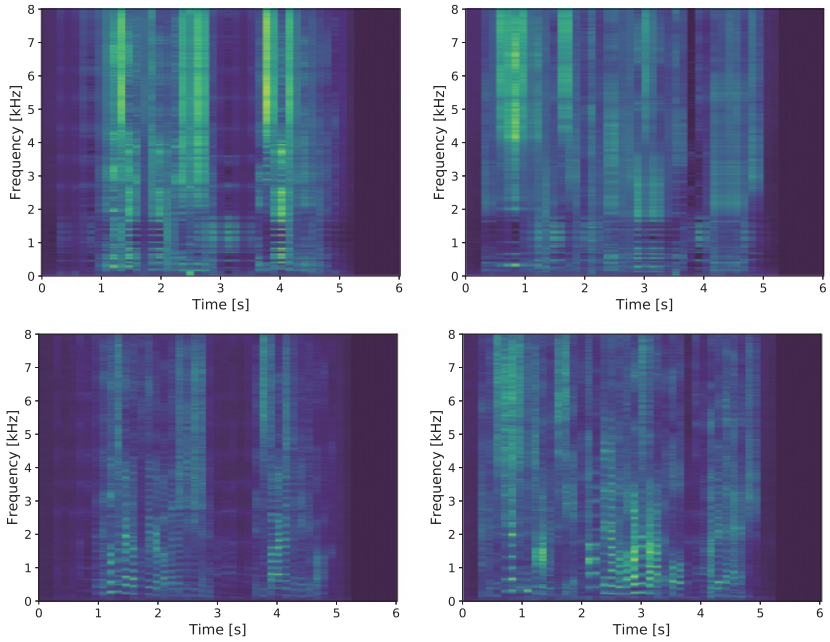

Figure 1: Example of the NMF models optimized using ILRMA (top) and the spectrograms of the corresponding source signals (bottom).

$$h_{j,k}(n) \leftarrow h_{j,k}(n) \sqrt{\frac{\sum_f |y_j(f,n)|^2 b_{j,k}(f)/v_j^2(f,n)}{\sum_f b_{j,k}(f)/v_j(f,n)}}, \tag{3.4}$$

where $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$.

One important feature of ILRMA is that the log likelihood, equation 2.9, is nondecreasing at each iteration of the algorithm and is shown experimentally to converge quickly. However, one limitation is that since $v_j(f,n)$ is restricted to equation 2.10, it can fail to work for sources with spectrograms that do not follow equation 2.10. Figure 1 shows an example of the NMF model optimally fitted to a speech spectrogram. As can be seen from this example, there is still plenty of room for improvement in the model design.

3.2 DNN Approach. As an alternative to the NMF model, some attempts have recently been made to combine deep neural networks (DNNs) with the LGM-based multichannel source separation framework (Nugraha et al., 2016; Mogami et al., 2018). Nugraha et al. (2016) and Mogami et al. (2018) propose algorithms where $v_j(f,n)$ is updated at each iteration to

the output of pretrained DNNs. In particular, with the method in Mogami et al. (2018), a different DNN is trained for each source, and the jth DNN is trained so that it produces only spectra related to source j in noisy input spectra,

$$\tilde{\mathbf{v}}_j(n) \leftarrow \mathrm{DNN}_j(\tilde{\mathbf{y}}_j(n)) \ (n = 1, \ldots, N), \tag{3.5}$$

where $\mathrm{DNN}_j(\cdot)$ indicates the output of the pretrained DNN for source j, $\tilde{\mathbf{y}}_j(n) = \{|y_j(f, n \pm n')|\}_{f,n'}$ denotes the magnitude spectra of the estimate of the jth separated signal around the nth time frame, and $\tilde{\mathbf{v}}_j(n) = \{\sqrt{v_j(f,n)}\}_f$. Equation 3.5 can thus be seen as a process of refining the magnitude spectra of the separated signals according to the training examples of the known sources.

While this approach is noteworthy in that it can exploit the benefits of the representation power of DNNs for source power spectrum modeling, one drawback is that updating $v_j(f, n)$ in this way does not guarantee an increase in the log likelihood.

3.3 Source Separation Using Deep Generative Models. It is worth noting that there have been some attempts to apply deep generative models, including VAEs (Kingma & Welling, 2014; Kingma et al., 2014), and generative adversarial networks (GANs; Goodfellow et al., 2014) to monaural speech enhancement and source separation (Bando et al., 2018; Subakan & Smaragdis, 2018; Leglaive et al., 2018). As far as we know, their applications to multichannel source separation had yet to be proposed when our preprint paper on this work (Kameoka et al., 2018) was first made publicly available. Recently, it has been brought to our attention that several papers on applications of VAEs to multichannel speech enhancement have subsequently been published by different authors (Sekiguchi, Bando, Yoshii, & Kawahara, 2018; Leglaive et al., 2019). These methods are designed to enhance the speech of a particular speaker by using a VAE to model the spectrogram of that speaker. Hence, one limitation of these methods is that we must know which speaker is present in a test mixture.

4 Proposed Method

To address the limitations and drawbacks of the conventional methods, this letter proposes a multichannel source separation method using CVAEs for source spectrogram modeling. We briefly review the idea behind the VAEs and CVAEs in section 4.1 and present the proposed source separation algorithm in section 4.2, which we call the multichannel CVAE (MCVAE) or, more simply, the multichannel VAE (MVAE).

4.1 Variational Autoencoder. Variational autoencoders (VAEs) (Kingma & Welling, 2014; Kingma et al., 2014) are stochastic neural network

models consisting of encoder and decoder networks. The encoder network generates a set of parameters for the conditional distribution $q_\phi(\mathbf{z}|\mathbf{s})$ of a latent space variable $\mathbf{z}$ given input data $\mathbf{s}$, whereas the decoder network generates a set of parameters for the conditional distribution $p_\theta(\mathbf{s}|\mathbf{z})$ of the data $\mathbf{s}$ given the latent space variable $\mathbf{z}$. Given a training data set $\mathcal{S} = \{\mathbf{s}_m\}_{m=1}^M$, VAEs learn the parameters of the entire network so that the encoder distribution $q_\phi(\mathbf{z}|\mathbf{s})$ becomes consistent with the posterior $p_\theta(\mathbf{z}|\mathbf{s}) \propto p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})$. By using Jensen's inequality, the log marginal distribution of the data $\mathbf{s}$ can be lower-bounded by

$$\begin{aligned}\log p_\theta(\mathbf{s}) &= \log \int q_\phi(\mathbf{z}|\mathbf{s}) \frac{p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})} d\mathbf{z} \\ &\geq \int q_\phi(\mathbf{z}|\mathbf{s}) \log \frac{p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})} d\mathbf{z} \\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{s}) \| p(\mathbf{z})], \end{aligned} \tag{4.1}$$

where the difference between the left- and right-hand sides of equation 4.1 is given by

$$\begin{aligned}\log p_\theta(\mathbf{s}) \; &- \int q_\phi(\mathbf{z}|\mathbf{s}) \log \frac{p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})} d\mathbf{z} \\ &= \int q_\phi(\mathbf{z}|\mathbf{s}) \log \frac{p_\theta(\mathbf{s})q_\phi(\mathbf{z}|\mathbf{s})}{p_\theta(\mathbf{s},\mathbf{z})} d\mathbf{z} = \int q_\phi(\mathbf{z}|\mathbf{s}) \log \frac{q_\phi(\mathbf{z}|\mathbf{s})}{p_\theta(\mathbf{z}|\mathbf{s})} d\mathbf{z}, \end{aligned} \tag{4.2}$$

which is equal to the Kullback-Leibler divergence between $q_\phi(\mathbf{z}|\mathbf{s})$ and $p_\theta(\mathbf{z}|\mathbf{s})$. Obviously this is minimized when

$$q_\phi(\mathbf{z}|\mathbf{s}) = p_\theta(\mathbf{z}|\mathbf{s}). \tag{4.3}$$

This means we can make $q_\phi(\mathbf{z}|\mathbf{s})$ and $p_\theta(\mathbf{z}|\mathbf{s}) \propto p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})$ consistent by maximizing the lower bound of equation 4.1. One typical way of modeling $q_\phi(\mathbf{z}|\mathbf{s})$, $p_\theta(\mathbf{s}|\mathbf{z})$ and $p(\mathbf{z})$ is to assume gaussian distributions

$$q_\phi(\mathbf{z}|\mathbf{s}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{s}), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\mathbf{s}))), \tag{4.4}$$

$$p_\theta(\mathbf{s}|\mathbf{z}) = \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_\theta(\mathbf{z}), \mathrm{diag}(\boldsymbol{\sigma}_\theta^2(\mathbf{z}))), \tag{4.5}$$

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}), \tag{4.6}$$

where $\boldsymbol{\mu}_\phi(\mathbf{s})$ and $\boldsymbol{\sigma}_\phi^2(\mathbf{s})$ are the outputs of an encoder network with parameter ϕ, and $\boldsymbol{\mu}_\theta(\mathbf{z})$ and $\boldsymbol{\sigma}_\theta^2(\mathbf{z})$ are the outputs of a decoder network with parameter θ. Here, it should be noted that to compute the first term of this objective function, we must compute the expectation with respect to

$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. Although this expectation cannot be expressed in an analytical form, we can compute it by using a Monte Carlo approximation. However, simply sampling $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x})$ does not work, since once $\mathbf{z}$ is sampled, it is no longer a function of ϕ, which makes it impossible to evaluate the gradient of the objective function with respect to ϕ. Fortunately, by using a reparameterization $\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I})$ where $\odot$ indicates the element-wise product, sampling $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x})$ can be replaced by sampling $\boldsymbol{\epsilon}$ from the distribution, which is independent of ϕ. This allows us to compute the gradient of the first term of the objective function with respect to ϕ by using a Monte Carlo approximation of the expectation $\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\cdot]$. This technique is called a reparameterization trick. By using this reparameterization, the first term of the lower bound can be written as

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s}|\mathbf{z})] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0},\mathbf{I})}\Bigg[-\frac{1}{2}\sum_n \log 2\pi[\boldsymbol{\sigma}_\theta^2(\boldsymbol{\mu}_\phi(\mathbf{s}) + \boldsymbol{\sigma}_\phi(\mathbf{s}) \odot \boldsymbol{\epsilon})]_n \\
&\qquad - \sum_n \frac{(s_n - [\boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\mathbf{s}) + \boldsymbol{\sigma}_\phi(\mathbf{s}) \odot \boldsymbol{\epsilon})]_n)^2}{2[\boldsymbol{\sigma}_\theta^2(\boldsymbol{\mu}_\phi(\mathbf{s}) + \boldsymbol{\sigma}_\phi(\mathbf{s}) \odot \boldsymbol{\epsilon})]_n} \Bigg],
\end{aligned} \tag{4.7}$$

where $[\cdot]_n$ denotes the nth element of a vector. We can confirm from equation 4.7 that the second term reduces to a negative weighted squared error between $\mathbf{s}$ and $\boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\mathbf{s}))$ when $\boldsymbol{\epsilon} = \mathbf{0}$, which can be interpreted as an autoencoder reconstruction error. On the other hand, the second term of equation 4.1 is given as the negative KL divergence between $q_\phi(\mathbf{z}|\mathbf{s})$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. This term can be interpreted as a regularization term that forces each element of the encoder output to be independent and normally distributed.

Conditional VAEs (CVAEs; Kingma et al., 2014) are an extended version of VAEs where the only difference is that the encoder and decoder networks can take an auxiliary variable c as an additional input. With CVAEs, equations 4.4 and 4.5 are replaced with

$$q_\phi(\mathbf{z}|\mathbf{s}, c) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{s}, c), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\mathbf{s}, c))), \tag{4.8}$$

$$p_\theta(\mathbf{s}|\mathbf{z}, c) = \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_\theta(\mathbf{z}, c), \mathrm{diag}(\boldsymbol{\sigma}_\theta^2(\mathbf{z}, c))), \tag{4.9}$$

and the variational lower bound to be maximized becomes

$$\mathcal{J}(\phi, \theta) = \mathbb{E}_{(\mathbf{s},c)\sim p_{\mathrm{D}}(\mathbf{s},c)}\big[\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{s},c)}[\log p(\mathbf{s}|\mathbf{z}, c)] - \mathrm{KL}[q(\mathbf{z}|\mathbf{s}, c)\|\, p(\mathbf{z})]\big], \tag{4.10}$$

where $\mathbb{E}_{(\mathbf{s},c)\sim p_{\mathrm{D}}(\mathbf{s},c)}[\cdot]$ denotes the sample mean over the training examples $\{\mathbf{s}_m, c_m\}_{m=1}^M$.

One notable feature of CVAEs is that they are able to learn a "disentangled" latent representation underlying the data of interest. For example, when a CVAE is trained using the MNIST data set of handwritten digits and c as the digit class label, $\mathbf{z}$ and c are disentangled so that $\mathbf{z}$ represents the factors of variation corresponding to handwriting styles. We can thus generate images of a desired digit with random handwriting styles from the trained decoder by specifying c and randomly sampling $\mathbf{z}$. Analogously, we would be able to obtain a generative model that can represent the spectrograms of a variety of sound sources if we could train a CVAE using class-labeled training examples.

4.2 Multichannel VAE. Let $\tilde{\mathbf{S}} = \{s(f,n)\}_{f,n}$ be the complex spectrogram of a particular sound source and c be the class label of that source. Here, we assume that a class label comprises one or more categories, each consisting of multiple classes. We thus represent c as a concatenation of one-hot vectors, each of which is filled with 1 at the index of a class in a certain category and with 0 everywhere else. For example, if we consider speaker identities as the only class category, c will be represented as a single one-hot vector, where each element is associated with a different speaker.

We now model the generative model of $\tilde{\mathbf{S}}$ using a CVAE with an auxiliary input c. So that the decoder distribution has the same form as the LGM, equation 2.6, we define it as a zero-mean complex gaussian distribution,

$$\begin{aligned} p_\theta(\tilde{\mathbf{S}}|\mathbf{z},c,g) &= \mathcal{N}_{\mathbb{C}}(\tilde{\mathbf{S}}|\mathbf{0}, g\cdot \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{z},c))) \\ &= \prod_{f,n} \mathcal{N}_{\mathbb{C}}(s(f,n)|0, g\cdot \sigma^2_\theta(f,n;\mathbf{z},c)), \end{aligned} \tag{4.11}$$

where $\sigma^2_\theta(f,n;\mathbf{z},c)$ denotes the (f,n)th element of the decoder output $\boldsymbol{\sigma}^2_\theta(\mathbf{z},c)$ and g represents the global scale of the generated spectrogram. For the encoder distribution $q_\phi(\mathbf{z}|\tilde{\mathbf{S}},c)$, we adopt a regular gaussian distribution

$$\begin{aligned} q_\phi(\mathbf{z}|\tilde{\mathbf{S}},c) &= \mathcal{N}_{\mathbb{C}}(\mathbf{z}|\boldsymbol{\mu}_\phi(\tilde{\mathbf{S}},c), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\tilde{\mathbf{S}},c))) \\ &= \prod_k \mathcal{N}(z(k)|\mu_\phi(k;\tilde{\mathbf{S}},c), \sigma^2_\phi(k;\tilde{\mathbf{S}},c)), \end{aligned} \tag{4.12}$$

where $z(k)$, $\mu_\phi(k;\tilde{\mathbf{S}},c)$, and $\sigma^2_\phi(k;\tilde{\mathbf{S}},c)$ represent the kth elements of the latent space variable $\mathbf{z}$ and the encoder outputs $\boldsymbol{\mu}_\phi(\tilde{\mathbf{S}},c)$ and $\boldsymbol{\sigma}^2_\phi(\tilde{\mathbf{S}},c)$, respectively. Given a set of labeled training examples $\{\tilde{\mathbf{S}}_m, c_m\}_{m=1}^M$, we train the decoder and encoder NN parameters θ and ϕ, respectively, prior to source separation, using the training objective

$$\mathcal{J}(\phi,\theta) = \mathbb{E}_{(\tilde{\mathbf{S}},c)\sim p_{\mathrm{D}}(\tilde{\mathbf{S}},c)}\big[\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\tilde{\mathbf{S}},c)}[\log p(\tilde{\mathbf{S}}|\mathbf{z},c)] - \mathrm{KL}[q(\mathbf{z}|\tilde{\mathbf{S}},c)\|p(\mathbf{z})]\big], \tag{4.13}$$

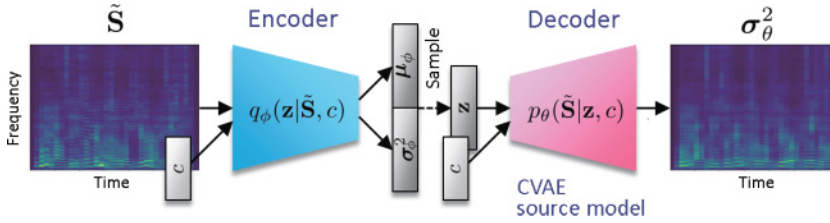

Figure 2: Illustration of the present CVAE.

where $\mathbb{E}_{(\tilde{\mathbf{S}},c)\sim p_{\mathrm{D}}(\tilde{\mathbf{S}},c)}[\cdot]$ denotes the sample mean over the training examples $\{\tilde{\mathbf{S}}_m, c_m\}_{m=1}^M$. Figure 2 shows the illustration of the present CVAE.

The trained decoder distribution $p_\theta(\tilde{\mathbf{S}}|\mathbf{z}, c, g)$ can be used as a universal generative model that is able to generate spectrograms of all the sources involved in the training examples where the latent space variable $\mathbf{z}$, the auxiliary input c, and the global scale g can be interpreted as the model parameters. According to the properties of CVAEs, we consider that the CVAE training promotes disentanglement between $\mathbf{z}$ and c where $\mathbf{z}$ characterizes the factors of intraclass variation, whereas c characterizes the factors of categorical variation that represent source identities. Estimating c from a test mixture corresponds to identifying which source is present in the mixture. There are, however, certain cases where we know which sources are present prior to separation. Thanks to the conditional modeling, we can also use our model in such cases by simply fixing c at a specified index. We call $p_\theta(\tilde{\mathbf{S}}|\mathbf{z}, c, g)$ the CVAE source model.

Since the CVAE source model is given in the same form as the LGM given by equation 2.6, we can develop a log likelihood that has the same expression as equation 2.9 if we use $p_\theta(\tilde{\mathbf{S}}_j|\mathbf{z}_j, c_j, g_j)$ to express the generative model of the complex spectrogram of source j where $v_j(f,n) = g_j \cdot \sigma_\theta^2(f,n;\mathbf{z}_j,c_j)$. Hence, we can search for a stationary point of the log likelihood by iteratively updating the separation matrices $\mathcal{W}$, the global scale parameter $\mathcal{G} = \{g_j\}_j$, and the VAE source model parameters $\Psi = \{\mathbf{z}_j, c_j\}_j$ so that the log likelihood is guaranteed to be nondecreasing at each iteration. We can use equation 3.1 and 3.2 to update $\mathcal{W}$, backpropagation to update Ψ, and

$$g_j \leftarrow \frac{1}{FN}\sum_{f,n}\frac{|y_j(f,n)|^2}{\sigma_\theta^2(f,n;\mathbf{z}_j,c_j)}, \tag{4.14}$$

to update $\mathcal{G}$ where $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$. Note that equation 4.14 maximizes equation 2.9 with respect to g_j when $\mathcal{W}$ and Ψ are fixed.

The proposed algorithm is thus summarized as algorithm 1. Note that we must take account of the sum-to-one constraints when updating c_j. This

Algorithm 1: MVAE Algorithm.

Train θ and ϕ using equation 4.13.

Initialize $\mathcal{W}$, $\mathcal{G}$ and $\Psi = \{\mathbf{z}_j, c_j\}_j$.

while until convergence **do**

for $j = 1$ to J **do**

Update $\mathbf{w}_j(0), \ldots, \mathbf{w}_j(F)$ using equations 3.1 and 3.2.

Update $\psi_j = \{\mathbf{z}_j, c_j\}$ using backpropagation.

Update g_j using equation 4.14.

end for

end while

can be easily implemented by inserting an appropriately designed softmax layer that outputs c_j,

$$c_j = \mathsf{softmax}(u_j), \tag{4.15}$$

and treating u_j as the parameter to be estimated instead.

The proposed MVAE is noteworthy in that it offers the advantages of the conventional methods concurrently: (1) it takes full advantage of the strong representation power of DNNs for source power spectrogram modeling, (2) the log likelihood is guaranteed to be nondecreasing at each iteration of the source separation algorithm, and (3) the criteria for CVAE training and source separation are consistent, thanks to the consistency between the expressions of the CVAE source model and the LGM. Figure 3 shows an example of the CVAE source model fitted to the speech spectrogram shown in Figure 1. We can confirm from this example that the CVAE source model is able to approximate the speech spectrogram somewhat better than the NMF model.

It is interesting to look at the differences between our method and the recently proposed VAE-based multichannel speech enhancement methods (Sekiguchi et al., 2018; Leglaive et al., 2019). The methods proposed in Sekiguchi et al. (2018) and Leglaive et al. (2019) model the spectrogram of a particular source to be enhanced using a regular VAE and express the spectrograms of the other sources using the NMF model. This allows these methods to handle semisupervised scenarios where interference sources are unseen in the training set. However, one limitation is that the target source to be enhanced must be specified prior to separation. With our method,

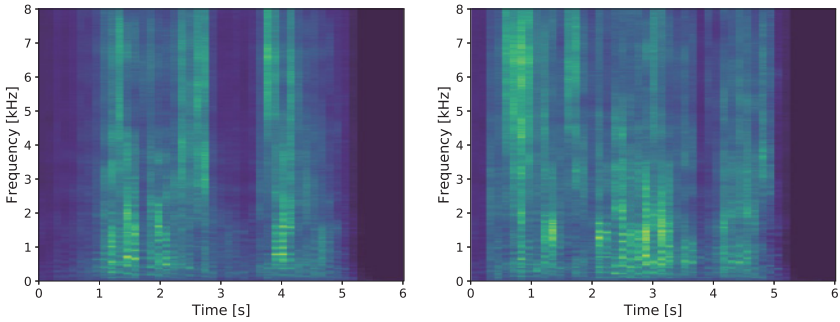

Figure 3: Example of the optimized CVAE source models corresponding to the source signals shown in Figure 1.

one limitation is that it can handle only supervised scenarios where audio samples of all the sources in a test mixture are included in the training set. However, if there is a sufficiently wide variety of sources in the training set, our method can be applied even without being informed about which of the sources in the training set are present in a test mixture. Our method can also be flexibly adapted to a scenario where we know which sources are present by simply specifying (instead of having it estimate) c_j, thanks to the conditional modeling. Another important feature of our model lies in its ability to capture the time-frequency interdependence in the STFT coefficients of each source thanks to the network design for the encoder and decoder, as presented in the section 4.3.

4.3 Network Architectures. We propose designing the encoder and decoder networks using fully convolutional architectures to allow the encoder to take a spectrogram as an input and allow the decoder to output a spectrogram of the same length instead of a single-frame spectrum. This allows the networks to capture time dependencies in spectral sequences. Although RNN-based architectures are a natural choice for modeling time series data, RNNs are unsuited to parallel implementations, and so both the training and conversion processes can be computationally demanding. Motivated by the recent success of sequential modeling using convolutional neural networks (CNNs) in the field of natural language processing (Dauphin, Fan, Auli, & Grangier, 2017) and the fact that CNNs are more suited to parallel implementations than RNNs, we use CNN-based architectures to design the encoder and decoder, as detailed below.

As detailed in Figure 4, we use 1D CNNs to design the encoder and the decoder networks by treating $\tilde{\mathbf{S}}$ as an image of size $1 \times N$ with F channels. Specifically, we use a gated CNN (Dauphin et al., 2017), which was originally introduced to model word sequences for language modeling and was shown to outperform long short-term memory (LSTM) language models

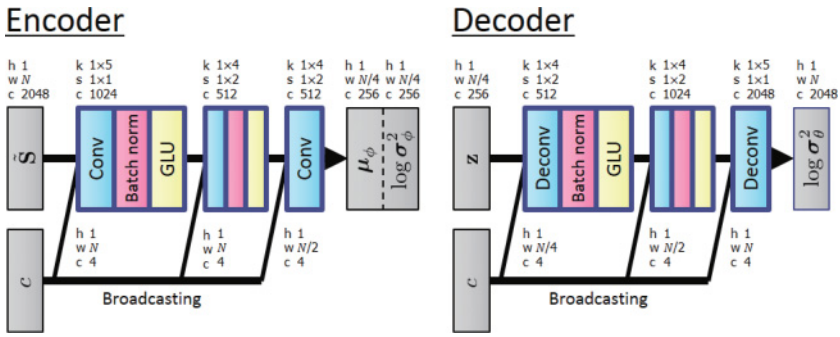

Figure 4: Network architectures of the encoder and decoder. Here, the inputs and outputs of the encoder and decoder are interpreted as images, where h, w, and c denote the height, width, and channel number, respectively. Conv, Batch norm, GLU, and Deconv denote convolution, batch normalization, gated linear unit, and transposed convolution layers, respectively. k, s, and c denote the kernel size, stride size, and output channel number of a convolution layer, respectively. c is assumed to be appended along the channel dimension to the output of the previous layer. Note that all the networks are fully convolutional with no fully connected layers, thus allowing inputs to have arbitrary lengths.

trained in a similar setting. We previously employed gated CNN architectures for voice conversion (Kaneko, Kameoka, Hiramatsu, & Kashino, 2017; Kaneko & Kameoka, 2017; Kameoka, Kaneko, Tanaka, & Hojo, 2018) and monaural audio source separation (Li & Kameoka, 2018), and have already confirmed their effectiveness. The output of the GLU block used in the present model is defined as

$$\mathsf{GLU}(\mathbf{X}) = \mathsf{B}_1(\mathsf{L}_1(\mathbf{X})) \odot \mathsf{sigmoid}(\mathsf{B}_2(\mathsf{L}_2(\mathbf{X}))), \tag{4.16}$$

where $\odot$ denotes elementwise multiplication, $\mathbf{X}$ is the layer input, L_1 and L_2 denote convolution layers, B_1 and B_2 denote batch normalization layers, and $\mathsf{sigmoid}$ denotes a sigmoid gate function. Similar to LSTMs, the output gate $\mathsf{sigmoid}(\mathsf{B}_2(\mathsf{L}_2(\mathbf{X})))$ multiplies each element of $\mathsf{B}_1(\mathsf{L}_1(\mathbf{X}))$ and controls what information should be propagated through the hierarchy of layers. At each GLU block in the encoder and decoder, a broadcast version of c is appended along the channel dimension to the output of the previous GLU block. The decoder network is devised in the same way as the encoder network with the only difference being that $\boldsymbol{\mu}_\theta = \mathbf{0}$. It should be noted that the entire architecture is fully convolutional with no fully connected layers. The trained decoder can therefore be used as a generative model of spectrograms with arbitrary lengths. This is particularly convenient when designing source separation systems since they can allow signals of any length.

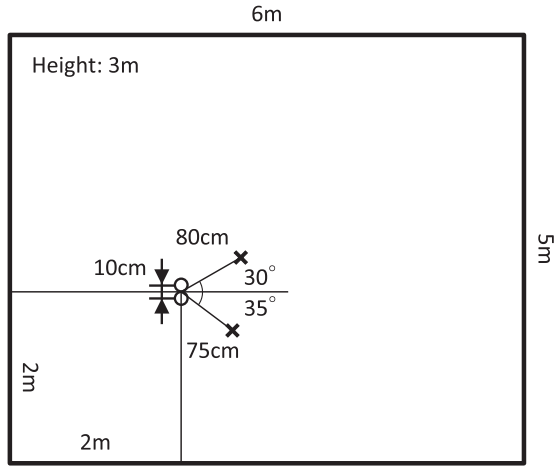

Figure 5: Simulated room configuration.

5 Experiments

5.1 Experimental Settings. To confirm the effect of the incorporation of the CVAE source model, we conducted experiments involving a supervised determined source separation task using speech mixtures. We excerpted speech utterances from the Voice Conversion Challenge (VCC) 2018 data set (Lorenzo-Trueba et al., 2018), which consists of recordings of six female and six male U.S. English speakers. Specifically, we used the utterances of two female speakers, SF1 and SF2, and two male speakers, SM1 and SM2, for CVAE training and source separation. We considered speaker identities as the only source class category. Thus, c was a four-dimensional one-hot vector. The audio files for each speaker were manually segmented into 116 short sentences (each about 7 minutes long) where 81 and 35 sentences (about 5 and 2 minutes long, respectively) were provided as training and evaluation sets, respectively.

We used two-channel recordings of two sources as the test data, which we synthesized using the simulated room impulse responses (RIRs) generated using the image method (Allen & Berkley, 1979) and the real RIRs measured in an anechoic room (ANE) and an echo room (E2A). Figure 5 shows the two-dimensional configuration of the room for obtaining the simulated RIRs. $\circ$ and $\times$ represent the positions of microphones and sources, respectively. The reverberation time (RT_{60}) (Schroeder, 1965) of the simulated RIRs could be controlled according to the setting of the reflection coefficient of the walls. To simulate anechoic and echoic environments, we created test signals with the reflection coefficients set at 0.20 and 0.80, respectively. The corresponding RT_{60}s were 78 ms and 351 ms, respectively. For the measured

RIRs, we used the data included in the RWCP Sound Scene Database in Real Acoustic Environments (Nakamura, Hiyane, Asano, & Endo, 1999). RT_{60} of the anechoic room (ANE) and the echo room (E2A) were 173 ms and 225 ms, respectively.

We generated 10 speech mixtures for each speaker pair, SF1 + SF2, SF1 + SM1, SM1 + SM2, and SF2 + SM2. Hence, there were 40 test signals for each recording condition, each of which was about 4 to 7 s long. All the speech signals were resampled at 16,000 Hz. The STFT frame length was set at 256 ms, and a Hamming window was used with an overlap length of 128 ms.

5.2 Baseline and Proposed Methods. We chose ILRMA (Kameoka et al., 2010; Kitamura et al., 2016, 2017) and the recently proposed DNN approach, the independent deeply learned matrix analysis (IDLMA; Mogami et al., 2018) as baseline methods for comparison. With ILRMA, we set K_j at 10 for all j. The IDLMA algorithm can be implemented by replacing the steps b) and c) in our algorithm with equation 3.5. Thus, ILRMA, IDLMA, and the proposed method differ only in the way $v_j(f, n)$ is modeled and estimated, and so the comparisons with the baseline methods would demonstrate the effect of our model. For a fair comparison, we used the same training data as those described in section 5.1 to train the DNN in equation 3.5. According to the settings in Mogami et al. (2018), we designed the DNN using four fully connected layers, each of which had 2048 units and was followed by a rectified linear unit (ReLU). The source separation algorithms were run for 40 iterations for the proposed method and 100 iterations for the baseline methods. Although the original ILRMA is a fully blind (unsupervised) approach, we also tested its supervised version for a fair comparison where the basis spectra were pretrained using the same training data. Specifically, we applied the NMF algorithm, which consisted of performing equations 3.3 and 3.4, to the audio samples of each source to obtain the basis spectra. We then constructed $\mathcal{B}$ by concatenating the obtained basis spectra of each source. Here we refer to the supervised version of ILRMA as sILRMA. For the proposed method, $\mathcal{W}$ was initialized using ILRMA run for 30 iterations, and Adam optimization (Kingma & Ba, 2015) was used for CVAE training and the estimation of Ψ in the source separation algorithm. The network configuration we used for the proposed method is shown in detail in Figure 4.

5.3 Results. To evaluate the source separation performance, we took the averages of the signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), and signal-to-artifact ratio (SAR; Vincent, Gribonval, & Févotte, 2006) of the separated signals obtained with the baseline and proposed methods using 10 test signals for each speaker pair. Figures 6 to 9 show the average SDRs, SIRs, and SARs obtained with the baseline and proposed methods under each recording condition. As the results show, the proposed method

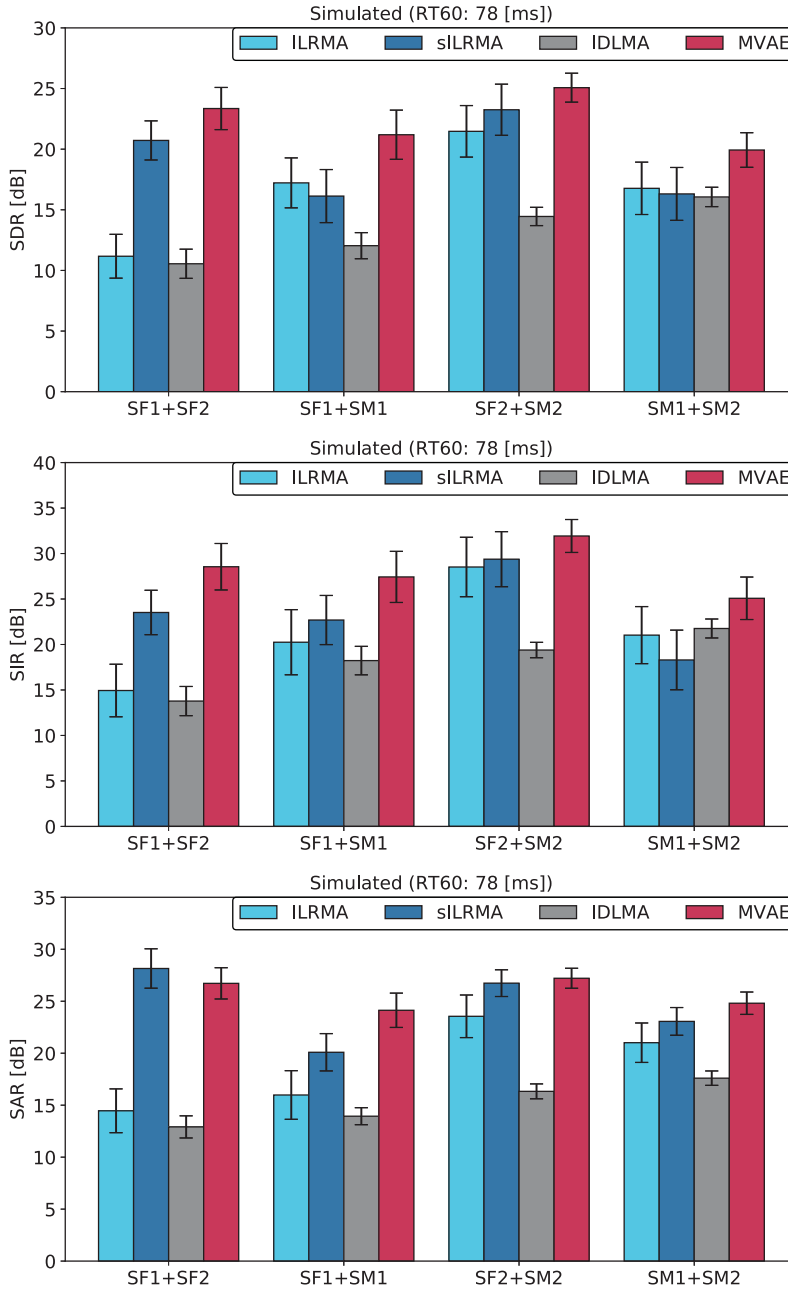

Figure 6: Average SDRs, SIRs, and SARs obtained with the baseline and proposed methods under a simulated recording condition with RT_{60} of 78 ms.

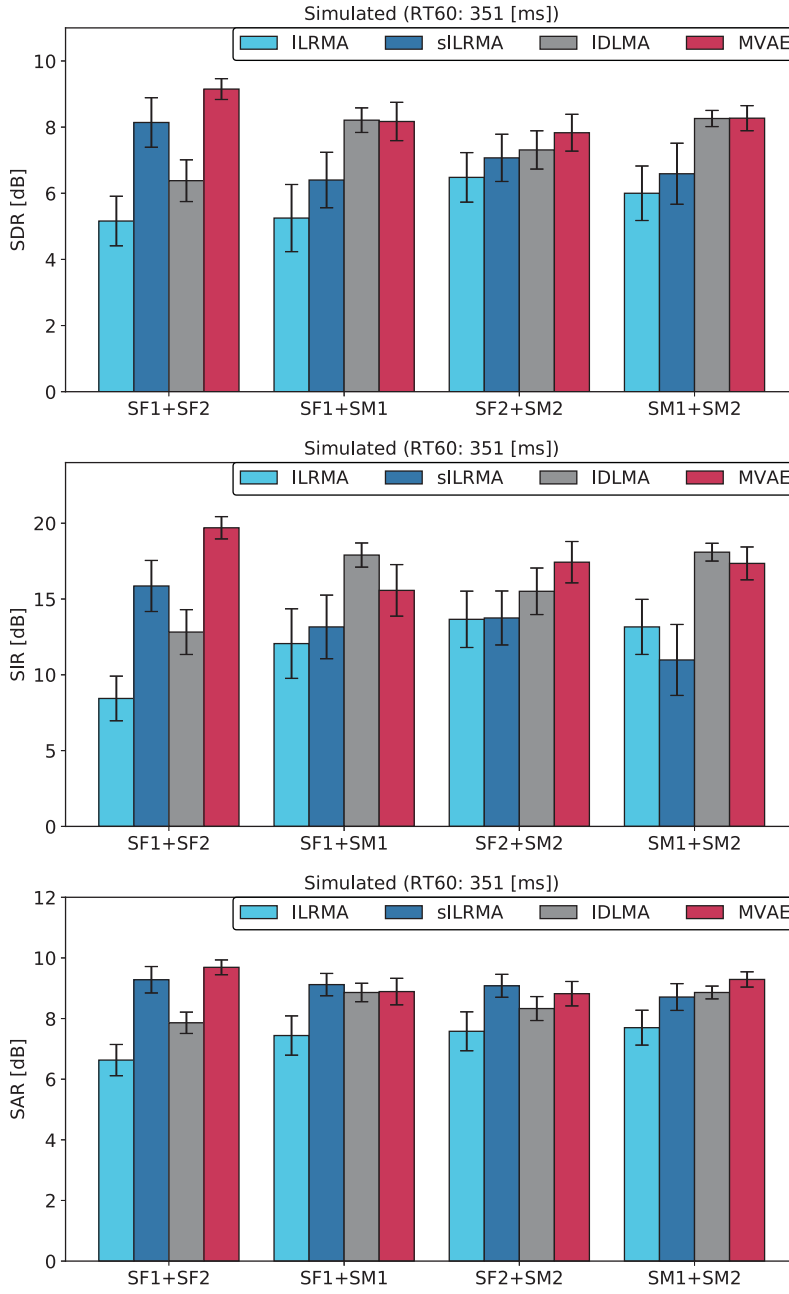

Figure 7: Average SDRs, SIRs, and SARs obtained with the baseline and proposed methods under a simulated recording condition with RT_{60} of 351 ms.

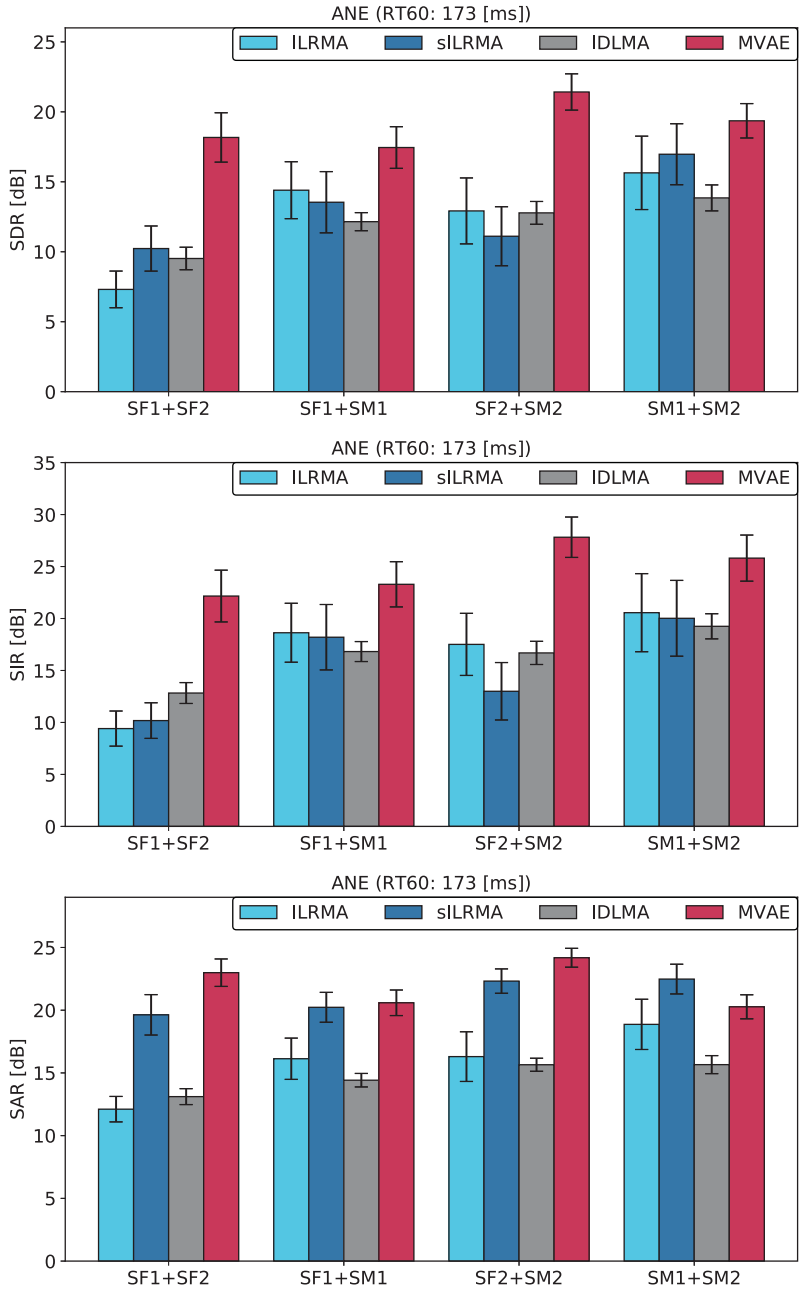

Figure 8: Average SDRs, SIRs, and SARs obtained with the baseline and proposed methods under the ANE recording condition with RT_{60} of 173 ms.

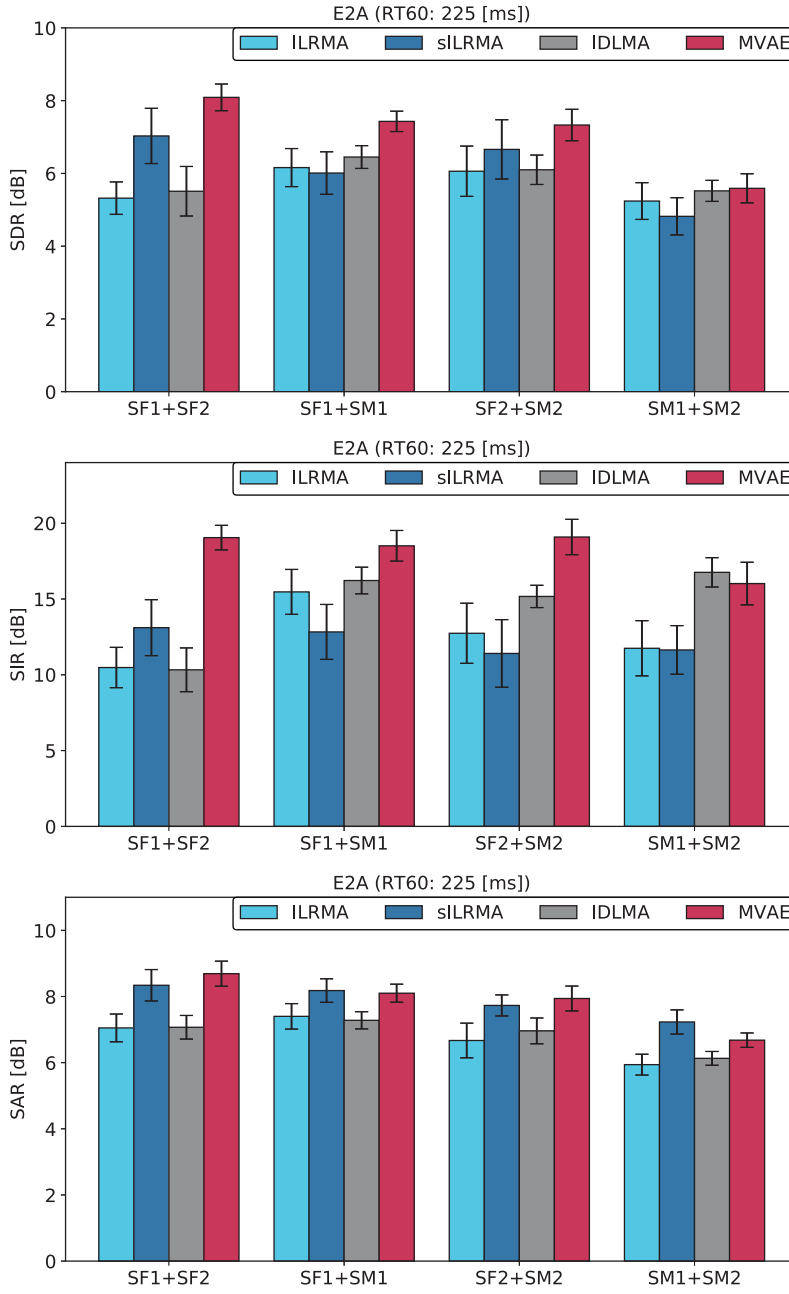

Figure 9: Average SDRs, SIRs, and SARs obtained with the baseline and proposed methods under the E2A recording condition with RT_{60} of 225 ms.

significantly outperformed the baseline methods for most of the test data in terms of SDR, revealing the advantage of the proposed approach. (Audio samples are provided at http://www.kecl.ntt.co.jp/people/kameoka.hirokazu/Demos/mvae-ass/.)

As can be seen from comparisons between the results in Figures 6 and 7 and those in Figures 8 and 9, there were noticeable performance degradations with both the baseline and proposed methods when the reverberation became relatively long. We have recently successfully incorporated the idea of jointly solving dereverberation and source separation (Kameoka et al., 2010; Yoshioka, Nakatani, Miyoshi, & Okuno, 2011; Kagami, Kameoka, & Yukawa, 2018) into the method to overcome these degradations (Inoue, Kameoka, Li, Seki, & Makino, 2019).

6 Conclusion

This letter proposed a multichannel source separation technique, the multichannel variational autoencoder (MVAE) method. The method used VAEs to model and estimate the power spectrograms of the sources in mixture signals. The key features of the MVAE are that (1) it takes full advantage of the strong representation power of deep neural networks for source power spectrogram modeling, (2) the log likelihood is guaranteed to be nondecreasing at each iteration of the source separation algorithm, and (3) the criteria for the VAE training and source separation are consistent, which contributed to obtaining better separations than with conventional methods. While the MVAE method was formulated under determined mixing conditions, it can be generalized so that it can also deal with underdetermined cases (Seki, Kameoka, Li, Toda, & Takeda, 2018).

Acknowledgments

This work was supported by JSPS KAKENHI 17H01763.

References

Allen, J. B., & Berkley, D. A. (1979). Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, *65*(4):943–950.

Bando, Y., Mimura, M., Itoyama, K., Yoshii, K., & Kawahara, T. (2018). Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 716–720). Piscataway, NJ: IEEE.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proc. International Conference on Machine Learning* (pp. 933–941).

Févotte, C., Bertin, N., & Durrieu, J.-L. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence. *Neural Computation*, *21*(3), 793–830.

Févotte, C., & Cardoso, J. F. (2005). Maximum likelihood approach for blind audio source separation using time-frequency gaussian models. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (pp. 78–81). Piscataway, NJ: IEEE.

Févotte, C., & Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, *23*(9), 2421–2456.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrance, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27* (pp. 2672–2680). Red Hook, NY: Curran.

Hiroe, A. (2006). Solution of permutation problem in frequency domain ICA using multivariate probability density functions. In J. Rosca, D. Erdogmus, J. C. Principe, & S. Haykin (Eds.), *Lecture Notes in Computer Science: vol. 3889. Independent Component Analysis and Blind Source Separation*. Berlin: Springer.

Inoue, S., Kameoka, H., Li, L., Seki, S., & Makino, S. (2019). Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 96–100). Piscataway, NJ: IEEE.

Kagami, H., Kameoka, H., & Yukawa, M. (2018). Joint separation and dereverberation of reverberant mixtures with determined multichannel non-negative matrix factorization. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 31–35). Piscataway, NJ: IEEE.

Kameoka, H., Goto, M., & Sagayama, S. (2006). *Selective amplifier of periodic and non-periodic components in concurrent audio signals with spectral control envelopes*. Technical Report 2006-MUS-66-13. Tokyo: Information Processing Society of Japan.

Kameoka, H., Kaneko, T., Tanaka, K., & Hojo, N. (2018). *StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks*. arXiv:1806.02169.

Kameoka, H., Li, L., Inoue, S., & Makino, S. (2018). *Semi-blind source separation with multichannel variational autoencoder*. arXiv:1808.00892.

Kameoka, H., Yoshioka, T., Hamamura, M., Le Roux, J., & Kashino, K. (2010). Statistical model of speech signals based on composite autoregressive system with application to blind source separation. In V. Vigeron, V. Zarzoso, E. Moreau, R. Gribonval, & E. Vincent (Eds.), *Lecture Notes in Computer Science: vol. 6365. Latent Variable Analysis and Signal Separation: LVAACA 2010*. Berlin: Springer.

Kaneko, T., & Kameoka, H. (2017). *Parallel-data-free voice conversion using cycle-consistent adversarial networks*. arXiv:1711.11293.

Kaneko, T., Kameoka, H., Hiramatsu, K., & Kashino, K. (2017). Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks. In *Proc. Annual Conference of the International Speech Communication Association* (pp. 1283–1287). Red Hook, NY: Curran.

Kim, T., Eltoft, T., & Lee, T.-W. (2006). Independent vector analysis: An extension of ICA to multivariate components. In J. Roscay D. Erdogmus, J. C. Principe, & S. Haykin (Eds.), *Lecture Notes in Computer Science: Vol. 3889. Independent Component Analysis and Blind Signal Separation: ICA 2006*. Berlin: Springer.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations*.

Kingma, D. P., Rezendey, D. J., Mohamedy, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27* (pp. 3581–3589). Red Hook, NY: Curran.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *Proc. International Conference on Learning Representations*.

Kitamura, D., Ono, N., Sawada, H., Kameoka, H., & Saruwatari, H. (2016). Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *24*(9), 1626–1641.

Kitamura, D., Ono, N., Sawada, H., Kameoka, H., & Saruwatari, H. (2017). Determined blind source separation with independent low-rank matrix analysis. In S. Makino (Ed.), *Audio source separation* (pp. 125–155). Berlin: Springer.

Leglaive, S., Girin, L., & Horaud, R. (2018). A variance modeling framework based on variational autoencoders for speech enhancement. In *Proc. IEEE International Workshop on Machine Learning for Signal Processing*. Piscataway, NJ: IEEE.

Leglaive, S., Girin, L., & Horaud, R. (2019). *Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization*. arXiv:1811.06713.

Li, L., & Kameoka, H. (2018). Deep clustering with gated convolutional networks. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 16–20). Piscataway, NJ: IEEE.

Lorenzo-Trueba, J., Yamagishi, J., Toda, T., Saito, D., Villavicencio, F., Kinnunen, T., & Ling, Z. (2018). *The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods*. arXiv:1804.04262.

Mogami, S., Sumino, H., Kitamura, D., Takamune, N., Takamichi, S., Saruwatari, H., & Ono, N. (2018). Independent deeply learned matrix analysis for multichannel audio source separation. In *Proc. European Signal Processing Conference*. Piscataway, NJ: IEEE.

Nakamura, S., Hiyane, K., Asano, F., & Endo, T. (1999). Sound scene data collection in real acoustical environments. *Journal of the Acoustical Society of Japan (E)*, *20*(3), 225–231.

Nakano, M., Kameoka, H., Le Roux, J., Ono, N., & Sagayama, S. (2010). Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with beta-divergence. In *Proc. IEEE International Workshop on Machine Learning for Signal Processing*. Piscataway, NJ: IEEE.

Nugraha, A. A., Liutkus, A., & Vincent, E. (2016). Multichannel audio source separation with deep neural networks. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *24*(9), 1652–1664.

Ono, N. (2011). Stable and fast update rules for independent vector analysis based on auxiliary function technique. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (pp. 189–192). Piscataway, NJ: IEEE.

Ozerov, A., & Févotte, C. (2010). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, *18*(3), 550–563.

Ozerov, A., & Kameoka, H. (2018). Gaussian model based multichannel separation. In E. Vincent, T. Virtanen, & S. G. (Eds.), *Audio source separation and speech enhancement*. Berlin: Springer.

Sawada, H., Kameoka, H., Araki, S., & Ueda, N. (2013). Multichannel extensions of non-negative matrix factorization with complex-valued data. *IEEE Transactions on Audio, Speech and Language Processing*, *21*(5), 971–982.

Schroeder, M. R. (1965). New method of measuring reverberation time. *Journal of the Acoustical Society of America*, *37*(3), 409–412.

Seki, S., Kameoka, H., Li, L., Toda, T., & Takeda, K. (2018). *Generalized multichannel variational autoencoder for underdetermined source separation*. arXiv:1810.00223.

Sekiguchi, K., Bando, Y., Yoshii, K., & Kawahara, T. (2018). Bayesian multichannel speech enhancement with a deep speech prior. In *Proc. Asia Pacific Signal and Information Processing Association Annual Summit and Conference* (pp. 1233–1239). Piscataway, NJ: IEEE.

Smaragdis, P. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (pp. 177–180). Piscataway, NJ: IEEE.

Subakan, Y., & Smaragdis, P. (2018). Generative adversarial source separation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 26–30). Piscataway, NJ: IEEE.

Vincent, E., Arberet, S., & Gribonval, R. (2009). Underdetermined instantaneous audio source separation via local gaussian modeling. In T. Adali, C. Jutten, J. M. T. Romana, & A. K. Barros (Eds.), *Lecture Notes in Computer Science: vol. 5441. Independent Component Analysis and Signal Separation: ICA 2009* (pp. 775–782). Berlin: Springer.

Vincent, E., Gribonval, R., & Févotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, *14*(4), 1462–1469.

Yoshioka, T., Nakatani, T., Miyoshi, M., & Okuno, H. G. (2011). Blind separation and dereverberation of speech mixtures by joint optimization. *IEEE Transactions on Audio, Speech and Language Processing*, *19*(1), 69–84.

Received February 1, 2019; accepted May 3, 2019.