IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 19, 2020, accepted December 4, 2020, date of publication December 18, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3045704

FastMVAE: A Fast Optimization Algorithm for the Multichannel Variational Autoencoder Method

LI LI[1], (Student Member, IEEE), HIROKAZU KAMEOKA[2], (Senior Member, IEEE), SHOTA INOUE[1], AND SHOJI MAKINO[1], (Fellow, IEEE)
[1]Graduate School of Systems and Information Engineering, University of Tsukuba, Ibaraki 3050821, Japan
[2]NTT Communication Science Laboratories, Kanagawa 2430198, Japan

Corresponding author: Li Li (lili@mmlab.cs.tsukuba.ac.jp)

This work was supported in part by the JSPS KAKENHI under Grant 18J20059, and in part by the JST CREST under Grant JPMJCR19A3.

ABSTRACT This paper proposes a fast optimization algorithm for the multichannel variational autoencoder (MVAE) method, a recently proposed powerful multichannel source separation technique. The MVAE method can achieve good source separation performance thanks to a convergence-guaranteed optimization algorithm and the idea of jointly performing multi-speaker separation and speaker identification. However, one drawback is the high computational cost of the optimization algorithm. To overcome this drawback, this paper proposes using an auxiliary classifier VAE, an information-theoretic extension of the conditional VAE (CVAE), to train the generative model of the source spectrograms and using it to efficiently update the parameters of the source spectrogram models at each iteration of the source separation algorithm. We call the proposed algorithm "FastMVAE" (or fMVAE for short). Experimental evaluations revealed that the proposed fast algorithm can achieve high source separation performance in both speaker-dependent and speaker-independent scenarios while significantly reducing the computational time compared to the original MVAE method by more than 90% on both GPU and CPU. However, there is still room for improvement of about 3 dB compared to the original MVAE method.

INDEX TERMS Multichannel source separation, multichannel variational autoencoder (MVAE) method, FastMVAE algorithm, auxiliary classifier VAE.

I. INTRODUCTION

Blind source separation (BSS), a technique for separating out individual source signals from microphone array inputs without any information about the sources or array geometry, has a wide range of applications, including hearing aids, automatic speech recognition, music editing, and music information retrieval.

In BSS, the frequency-domain approach is usually preferred since it enables a fast implementation compared with the time-domain approach. It is also notable in that it provides the flexibility of allowing us to utilize various models for the time-frequency (TF) representations of source signals. One example of this approach involves independent vector analysis (IVA) [1], [2], which achieves frequency-wise source separation and permutation alignment simultaneously by assuming the magnitudes of the frequency components originating from the same source vary coherently over time.

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Wang.

Multichannel extensions of non-negative matrix factorization (NMF), e.g., multichannel NMF (MNMF) [3], [4] and independent low-rank matrix analysis (ILRMA) [5]–[7], adopt the NMF concept for source spectrogram modeling in order to make use of the spectro-temporal structure underlying each source as a clue to separation. Specifically, the power spectrogram of each source signal is approximated as the linear sum of a limited number of basis spectra scaled by time-varying amplitudes. Owing to the fact that ILRMA reduces to IVA when it has only one flat basis spectrum, it can be seen that ILRMA has more flexibility than IVA in capturing the spectro-temporal structure in each source [6]. Each of these methods is designed to solve an inverse problem of estimating source signals based on a generative model of mixture signals. In this sense, these methods are categorized as a generative approach.

Meanwhile, given the recent advances achieved by deep neural network (DNN)-based speaker separation methods, including deep clustering (DC) [8], [9] and permutation invariant training (PIT) [10], [11], a discriminative approach

has recently proved powerful in monaural source separation tasks, including both speaker-dependent and -independent scenarios [12]–[15]. The general idea is to train a DNN that predicts TF masks or TF embeddings from a given mixture signal based on spectro-temporal features. When multiple microphones are available, spatial information can be utilized to improve separation performance [16]. Although these methods can achieve reasonably good separation, the TF masking process can cause unwanted distortion or musical noise in the separated speech. To avoid distortion and artificial noise and fully exploit the benefits of multichannel inputs, some efforts have been made to integrate DNNs into traditional microphone array processing frameworks such as beamforming [17], [18].

The success of these single-channel DNN-based methods attests to the excellent ability of DNNs to capture and learn the structure of spectrograms. Recently, some attempts have also been made to incorporate DNNs into the generative approach mentioned earlier [19]–[23]. As an example, a multichannel source separation method using a conditional variational autoencoder (multichannel VAE or MVAE for short) [21] has been proposed with notable success in supervised determined source separation tasks. With the MVAE method, a conditional VAE (CVAE) [24] is trained using the spectrograms of clean speech samples along with the corresponding speaker ID as a conditioning class variable. This is done so that the trained decoder distribution can be used as a generative model of signals produced by all the sources included in a given training set, where the latent space variables and the class variables are the parameters to be estimated from an input mixture signal. This generative model is called the *CVAE source model*. At the separation phase, the MVAE algorithm iteratively updates the demixing matrix using the iterative projection (IP) method [25] and the underlying parameters of the CVAE source model using a gradient descent method, where the gradients of the parameters are computed by backpropagation. The separated signals can then be obtained by applying the estimated demixing matrix to the observed mixture signals. One important feature of this method is that it is designed to jointly perform multi-speaker separation and speaker identification.[1] This is particularly reasonable since these two tasks are interdependent in the sense that the solution to one task can help find the solution to the other task. Another advantage worth noting is that by using a carefully chosen step size or applying a backtracking line search, the model parameters can be updated so as not to decrease the log-likelihood at each iteration of the algorithm. However, one downside is the high computational cost of the backpropagation process involved in each iteration.

To address this drawback, this paper proposes an accelerated version of the MVAE algorithm called the "FastMVAE (or fMVAE)" algorithm. The idea is to use an auxiliary classifier VAE (ACVAE) [26], an information-theoretic extension of a CVAE, to pretrain the generative distribution of source spectrograms. An ACVAE consists of decoder, encoder, and classifier networks. These three networks are trained simultaneously so that the decoder can be used as a generative model of spectrograms conditioned on a speaker ID, and the encoder and classifier can be used to infer the latent variables characterizing the generative model and the speaker ID from a spectrogram input. Since the backpropagation process involved in the original MVAE algorithm can be replaced with the forward propagation of the trained encoder and classifier networks, the entire algorithm can be made extremely efficient. It should be noted that this paper is an extended full-paper version of our conference paper [27]. The additional contributions in comparison to [27] are as follows:

- To stabilize the parameter inference process of the fMVAE algorithm, especially in speaker-independent conditions, we propose an improved version of the fast algorithm based on a Product-of-Experts (PoE) framework [28] and evaluate the impact of different hyperparameter settings.
- We demonstrate the capability of the MVAE and fMVAE algorithms to handle speaker-independent scenarios by sufficiently increasing the variety of speakers and the number of samples in the training dataset.

The rest of this paper is structured as follows. After describing the formulation of the determined multichannel BSS problem and the MVAE method in Section II, we review related work in Section III. In Section IV, we present the core idea of the fMVAE algorithm along with the ACVAE concept and describe the details of the proposed algorithm. We demonstrate the effectiveness of the proposed method in both speaker-dependent and speaker-independent source separation tasks in Section V. Finally, we conclude the paper in Section VI.

II. MVAE METHOD

A. PROBLEM FORMULATION

Let us consider a determined situation where I source signals are captured by I microphones. Let $x_i(f,n)$ and $s_j(f,n)$ denote the short-time Fourier transform (STFT) coefficients of the signal observed at the ith microphone and the jth source signal, where f and n are the frequency and time indices, respectively. We denote the vectors containing $x_1(f,n),\ldots,x_I(f,n)$ and $s_1(f,n),\ldots,s_I(f,n)$ by

$$\mathbf{x}(f,n) = [x_1(f,n),\ldots,x_I(f,n)]^{\mathsf{T}} \in \mathbb{C}^I, \tag{1}$$

$$\mathbf{s}(f,n) = [s_1(f,n),\ldots,s_I(f,n)]^{\mathsf{T}} \in \mathbb{C}^I, \tag{2}$$

where $(\cdot)^{\mathsf{T}}$ denotes transpose. In a determined situation, the relationship between observed signals and source signals can be described as

$$\mathbf{s}(f,n) = \mathbf{W}^{\mathsf{H}}(f)\mathbf{x}(f,n), \tag{3}$$

$$\mathbf{W}(f) = [\mathbf{w}_1(f),\ldots,\mathbf{w}_I(f)] \in \mathbb{C}^{I\times I}, \tag{4}$$

[1] When applied to other tasks such as music source separation and speech enhancement, it should be rephrased as musical instrument identification and speech/noise classification, respectively.

where $\mathbf{W}^{\mathsf{H}}(f)$ is called the demixing matrix and $(\cdot)^{\mathsf{H}}$ denotes the Hermitian transpose. The aim of BSS methods is to estimate $\mathcal{W} = \{\mathbf{W}(f)\}_f$ solely from the observation $\mathcal{X} = \{\mathbf{x}(f,n)\}_{f,n}$.

In the following, we assume each source signal follows the local Gaussian model (LGM) [29], [30]. Namely, $s_j(f,n)$ is assumed to independently follow a zero-mean complex proper Gaussian distribution with variance (power spectral density) $v_j(f,n)$:

$$p(s_j(f,n)|v_j(f,n)) = \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, v_j(f,n)), \tag{5}$$

where $v_j(f,n) = \mathbb{E}[|s_j(f,n)|^2]$. If $s_j(f,n)$ and $s_{j'}(f,n)$ $(j \neq j')$ are independent, the density of $\mathbf{s}(f,n)$ becomes

$$\begin{aligned} p(\mathbf{s}(f,n)|\mathbf{V}(f,n)) &= \prod_j p(s_j(f,n)|v_j(f,n)) \\ &= \mathcal{N}_{\mathbb{C}}(\mathbf{s}(f,n)|\mathbf{0}, \mathbf{V}(f,n)), \end{aligned} \tag{6}$$

where $\mathbf{V}(f,n) = \mathrm{diag}[v_1(f,n), \ldots, v_I(f,n)]$. From (3) and (6), the density of $\mathbf{x}(f,n)$ is obtained as

$$\begin{aligned} &p(\mathbf{x}(f,n)|\mathbf{W}(f), \mathbf{V}(f,n)) \\ &\quad = |\mathbf{W}^{\mathsf{H}}(f)|^2 p(\mathbf{s}(f,n) = \mathbf{W}^{\mathsf{H}}(f)\mathbf{x}(f,n)|\mathbf{V}(f,n)), \end{aligned} \tag{7}$$

where $|\mathbf{W}^{\mathsf{H}}(f)|^2$ is the Jacobian of the mapping $\mathbf{x}(f,n) \mapsto \mathbf{s}(f,n)$. Hence, the log-likelihood of the separation matrices $\mathcal{W} = \{\mathbf{W}(f)\}_f$ and source model parameters $\mathcal{V} = \{v_j(f,n)\}_{f,n,j}$ given the observed mixture signals $\mathcal{X} = \{\mathbf{x}(f,n)\}_{f,n}$ is given by

$$\begin{aligned} &\log p(\mathcal{X}|\mathcal{W}, \mathcal{V}) \\ &= 2N \sum_f \log|\det \mathbf{W}^{\mathsf{H}}(f)| + \sum_j \log p(\boldsymbol{S}_j|\boldsymbol{V}_j) \\ &=^c 2N \sum_f \log|\det \mathbf{W}^{\mathsf{H}}(f)| \\ &\quad - \sum_{f,n,j} \left(\log v_j(f,n) + \frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{v_j(f,n)} \right), \end{aligned} \tag{8}$$

where we have used $=^c$ to denote equality up to constant terms, and a bold italic font to indicate a set consisting of TF elements, namely, $\boldsymbol{S}_j = \{s_j(f,n)\}_{f,n}$, and $\boldsymbol{V}_j = \{v_j(f,n)\}_{f,n}$. Note that (8) will be split into frequency-wise source separation problems if there is no additional constraint imposed on $v_j(f,n)$. This indicates that there is a permutation ambiguity in the separated components for each frequency. Thus, we usually need to group together the separated components of different frequency bins that originate from the same source after or during source separation. This process is called permutation alignment.

B. CVAE MODEL

One efficient way to eliminate the permutation ambiguity is to incorporate a constraint into $v_j(f,n)$ so that the spectral structures of sources can be utilized as a clue to the estimation of $\mathcal{W}$. The idea of the MVAE method is to use a CVAE [24] conditioned on a class variable $\mathbf{c}$ to model the complex spectrograms $\boldsymbol{S} = \{s(f,n)\}_{f,n}$ of source signals. Here, $\mathbf{c}$ is a

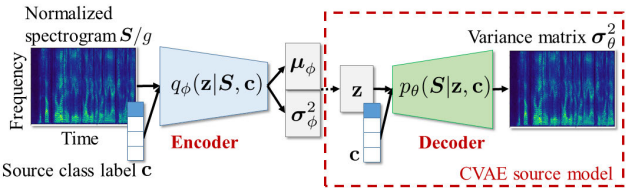

FIGURE 1. Illustration of CVAE used in MVAE.

one-hot vector consisting of C elements, indicating to which class the spectrogram $\boldsymbol{S}$ belongs. For example, if we consider speaker IDs as the class category, each element of $\mathbf{c}$ will be associated with a different speaker, and $\mathbf{c}$ will be filled with 1 at the index of a certain speaker and with 0 everywhere else.

A VAE is a stochastic neural network model consisting of an encoder and decoder, and a CVAE is an extended version that allows the encoder and decoder to include a conditioning class variable. In a CVAE, the decoder is modeled as a neural network (decoder network) that produces a set of parameters for a conditional distribution $p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$ of data $\boldsymbol{S}$ given a latent space variable $\mathbf{z}$ and a class variable $\mathbf{c}$, where θ denotes the network parameters. Figure 1 shows an illustration of CVAE. Computing the exact posterior $p_\theta(\mathbf{z}|\boldsymbol{S},\mathbf{c}) = p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})p(\mathbf{z})/p_\theta(\boldsymbol{S}|\mathbf{c})$ of $\mathbf{z}$ given $\boldsymbol{S}$ and $\mathbf{c}$ is usually very difficult since $p_\theta(\boldsymbol{S}|\mathbf{c})$ involves an intractable integral over $\mathbf{z}$. The idea of CVAEs is to sidestep the direct computation of this posterior by introducing another neural network (encoder network) for approximating the exact posterior $p_\theta(\mathbf{z}|\boldsymbol{S},\mathbf{c})$. As with the decoder network, the encoder network generates a set of parameters for the conditional distribution $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$, where ϕ denotes the network parameters. The goal is to learn the parameters of the encoder and decoder networks so that the encoder distribution $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$ becomes consistent with the posterior $p_\theta(\mathbf{z}|\boldsymbol{S},\mathbf{c}) \propto p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})p(\mathbf{z})$. Specifically, we train the encoder and decoder networks so that $\mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})||p_\theta(\mathbf{z}|\boldsymbol{S},\mathbf{c})]$ is minimized given M class-labeled training samples $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$. Since $\mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})||p_\theta(\mathbf{z}|\boldsymbol{S},\mathbf{c})] = \log p(\boldsymbol{S}) + \mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})||p(\mathbf{z}))] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})} \log p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$, this process amounts to minimizing

$$\begin{aligned} \mathcal{J}(\phi,\theta) = \mathbb{E}_{(\boldsymbol{S},\mathbf{c})\sim p_D(\boldsymbol{S},\mathbf{c})}\big[&\mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})||p(\mathbf{z})] \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})} \log p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})\big], \end{aligned} \tag{9}$$

where $\mathbb{E}_{(\boldsymbol{S},\mathbf{c})\sim p_D(\boldsymbol{S},\mathbf{c})}[\cdot]$ can be approximated as the sample mean over $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$, and $\mathrm{KL}[\cdot||\cdot]$ denotes the Kullback-Leibler divergence. While $p_D(\boldsymbol{S},\mathbf{c})$ can be approximated as the empirical distribution of $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$, $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$, $p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$ and $p(\mathbf{z})$ are distributions that need to be modeled.

In the MVAE method, a CVAE is used to model the entire complex spectrogram $\boldsymbol{S}$ of an utterance, conditioned on a speaker ID vector $\mathbf{c}$. $p(\mathbf{z})$ and $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$ are described as Gaussian distributions as with a regular CVAE:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}), \tag{10}$$

Algorithm 1 MVAE Algorithm

Require: Network parameter θ trained using (9), observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$

randomly initialize $\mathcal{W}$, Ψ
optional: update $\mathcal{W}$ using a BSS method
for $\ell = 1$ to $\mathscr{L}$ **do**
  for each source j of J **do**
    $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
    (updating source model parameters)
    initialize g_j using (19)
    normalization: $\tilde{\boldsymbol{S}}_j = \{y_j(f,n)/g_j\}_{f,n}$
    for $k = 1$ to 100 **do**
      update $\mathbf{z}_j$ and $\mathbf{c}_j$ using backpropagation
      while keeping θ fixed
    end for
    calculate $\sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    update g_j using (19)
    compute $v_j(f,n) = g_j \cdot \sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    (updating demixing matrices)
    update $\mathbf{w}_j(f)$ using the IP method (17), (18)
  end for
end for

$$q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\boldsymbol{S},\mathbf{c}), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\boldsymbol{S},\mathbf{c}))),$$
$$= \prod_k \mathcal{N}(z(k)|\mu_\phi(k;\boldsymbol{S},\mathbf{c}), \sigma_\phi^2(k;\boldsymbol{S},\mathbf{c})), \quad (11)$$

where $z(k)$, $\mu_\phi(k;\boldsymbol{S},\mathbf{c})$, and $\sigma_\phi^2(k;\boldsymbol{S},\mathbf{c})$ denote the kth element of the latent space variable $\mathbf{z}$ and the encoder outputs $\boldsymbol{\mu}_\phi(\boldsymbol{S},\mathbf{c})$ and $\boldsymbol{\sigma}_\phi^2(\boldsymbol{S},\mathbf{c})$, respectively. $p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$ is defined as a zero-mean complex proper Gaussian distribution with the same form as the LGM:

$$p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j) = \textstyle\prod_{f,n} \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, \sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)), \quad (12)$$

where $\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)$ denotes the (f,n)th element of the decoder output. Once the parameters θ and ϕ of the encoder and decoder are trained using speaker-labeled training utterances, the decoder with fixed θ can be used as a generative model of spectrograms for each speaker at test time.

Normalizing the mean and variance of each training sample is one of the common practices in neural network training. Similarly, in the CVAE training in the MVAE method, the total energy of each training utterance is normalized to 1. However, of course, the total energy of the spectrogram of each source in a test mixture can vary from source to source and does not necessarily equal 1. So that the generative model can flexibly bridge this gap, a scale parameter g is additionally incorporated into (12) and treated as a free parameter to be estimated at test time. Namely, the generative model of the complex spectrograms $\boldsymbol{S}_j$ of utterances of speaker j can be expressed as

$$p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j) = \textstyle\prod_{f,n} p_\theta(s_j(f,n)|\mathbf{z}_j,\mathbf{c}_j,g_j), \quad (13)$$

where

$$p_\theta(s_j(f,n)|\mathbf{z}_j,\mathbf{c}_j,g_j) = \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)), \quad (14)$$

and $\mathbf{z}_j$, $\mathbf{c}_j$, and g_j are the unknown parameters to be estimated. (13) is called the *CVAE source model*. We can immediately confirm that the decoder distribution in (12) corresponds to a particular case of (13) where $g_j = 1$. Since the CVAE source model is given in the same form as the LGM in (5), where $v_j(f,n)$ is given by $g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)$, using it as the generative model of each source leads to the same form of the log-likelihood as (8):

$$\begin{aligned}
&\log p(\mathcal{X}|\mathcal{W},\Psi,\mathcal{G}) \\
&= 2N\sum_f \log|\det \mathbf{W}^{\mathsf{H}}(f)| + \sum_j \log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j) \\
&=^c 2N\sum_f \log|\det \mathbf{W}^{\mathsf{H}}(f)| \\
&\quad - \sum_{f,n,j}\left(\log g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j) + \frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)}\right),
\end{aligned} \quad (15)$$

where $\mathcal{G} = \{g_j\}_j$ and $\Psi = \{\mathbf{z}_j,\mathbf{c}_j\}_j$.

Since $\mathbf{z}$ is assumed to follow $\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I})$ when θ and ϕ are trained, it would be reasonable to assume it as a prior distribution for $\mathbf{z}$ also at test time. The prior $p(\mathbf{c})$ is the empirical distribution of the training examples $\{\mathbf{c}_m\}_m$, expressed as a multinomial distribution. Thus, the log-posterior

$$\log p(\mathcal{X}|\mathcal{W},\Psi,\mathcal{G};\theta) + \log p(\mathbf{z}) + \log p(\mathbf{c}) \quad (16)$$

is the objective function to be maximized with respect to $\mathcal{W}$, Ψ, and $\mathcal{G}$. A stationary point of (16) can be found by iteratively updating these parameters so that (16) is guaranteed to be non-decreasing. To update $\mathcal{W}$, the following update rules, called the iterative projection (IP) [25], can be used:

$$\mathbf{w}_j \leftarrow (\mathbf{W}^{\mathsf{H}}(f)\boldsymbol{\Sigma}_j(f))^{-1}\mathbf{e}_j, \quad (17)$$

$$\mathbf{w}_j \leftarrow \frac{\mathbf{w}_j(f)}{\mathbf{w}_j^{\mathsf{H}}(f)\boldsymbol{\Sigma}_j(f)\mathbf{w}_j(f)}, \quad (18)$$

where $\boldsymbol{\Sigma}_j(f) = \frac{1}{N}\sum_n \mathbf{x}(f,n)\mathbf{x}^{\mathsf{H}}(f,n)/v_j(f,n)$ and $\mathbf{e}_j$ denotes the jth column of an $I \times I$ identity matrix. To update $\mathcal{G}$, the following update rule can be used:

$$g_j \leftarrow \frac{1}{FN}\sum_{f,n}\frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)}. \quad (19)$$

Note that (19) maximizes (16) with respect to g_j when $\mathcal{W}$ and Ψ are fixed. While keeping $\mathcal{W}$ and $\mathcal{G}$ fixed, a gradient descent method can be used to search for the optimal $\mathbf{z}_j$ and $\mathbf{c}_j$ that maximize (16), or equivalently $\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j) + \log p(\mathbf{z}_j) + \log p(\mathbf{c}_j)$ for each j in parallel, where each element of $\boldsymbol{S}_j$ is given by $s_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$. Note that estimating $\mathbf{c}_j$ from a test mixture corresponds to identifying which speaker is present in the mixture signal. When updating $\mathbf{c}_j$, the sum-to-one constraint must be taken into account. This is easily implemented by inserting an appropriately designed softmax layer that outputs $\mathbf{c}_j$,

$$\mathbf{c}_j = \mathrm{softmax}(\mathbf{u}_j), \quad (20)$$

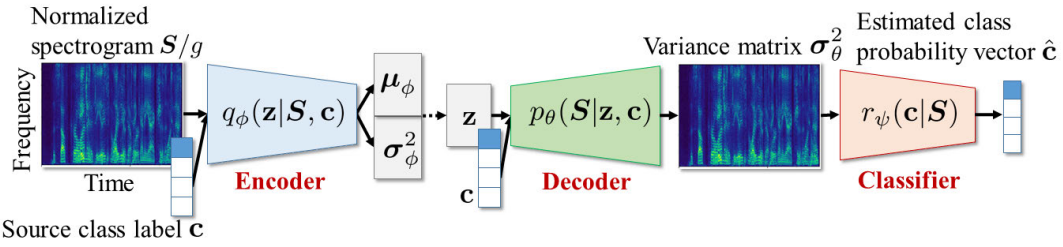

FIGURE 2. Illustration of ACVAE used in fMVAE.

and treating $\mathbf{u}_j$ as the parameter to be estimated instead. The source separation algorithm of the MVAE method is summarized in *Algorithm 1*. This algorithm is noteworthy that, if it is implemented appropriately, the log-likelihood of the model parameters is guaranteed to be non-decreasing at each iteration.

III. RELATED WORK

A. ILRMA

Another reasonable way of constraining power spectrograms involves employing the NMF model [31]. The NMF model expresses $v_j(f,n)$ as a linear sum of spectral templates $b_{j,1}(f),\ldots,b_{j,t}(f),\ldots,b_{j,T_j}(f) \geq 0$ scaled by time-varying magnitudes $h_{j,1}(n),\ldots,h_{j,t}(n),\ldots,h_{j,T_j}(n) \geq 0$:

$$v_j(f,n) = \sum_t^{T_j} b_{j,t}(f) h_{j,t}(n). \tag{21}$$

Note that a particular case where $T_j = 1$ and $b_{j,t}(f) = 1$ for all j is equivalent to assuming the norm $r_j(n) = \sqrt{\sum_f |s_j(f,n)|^2}$ follows a complex Gaussian distribution with time-varying variance $h_j(n)$. This is analogous to the assumption in IVA that the magnitudes of the STFT coefficients in all frequency bands originating from the same source tend to vary coherently over time [32].

The optimization algorithm of ILRMA consists of iteratively updating the demixing matrices $\mathcal{W}$ using the IP method, the basis templates $\mathcal{B} = \{b_{j,t}(f)\}_{f,j,t}$, and the activation matrix $\mathcal{H} = \{h_{j,t}(n)\}_{n,j,t}$ so that (8) is guaranteed to be non-decreasing at each iteration. To update $\mathcal{B}$ and $\mathcal{H}$, we can use the majorization-minimization (MM) algorithm [33]. The MM-based update rules can be derived as

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

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

B. DNN-BASED METHODS

Some attempts have recently been made to incorporate DNNs into the LGM-based multichannel source separation framework [19], [20]. With these methods, $v_j(f,n)$ is updated at each iteration as the output of pretrained DNNs. Independent deeply low-rank matrix analysis (IDLMA) [20] is a method designed to train a DNN for each source so that the jth DNN produces spectra related to source j when noisy spectra of the jth source are given as the input. Thus, each DNN can be seen as a source-dependent noise reduction system. One drawback of IDLMA is that updating $v_j(f,n)$ in this way does not guarantee an increase in the log-likelihood. Another drawback would be that it can perform poorly in speaker-independent scenarios.

C. VAE-BASED METHODS

Recently, deep generative models such as VAEs and generative adversarial networks (GANs) have proved powerful in source separation tasks [22], [23], [27], [34]–[40]. The idea of using a VAE to model the spectrum within each short-term frame was first proposed for single-channel speech enhancement [22]. This method, called VAE-NMF, enables speech enhancement in a semi-supervised manner by using a VAE to model the spectrogram of a target speaker and an NMF model to express unseen noise spectrograms. In this method, the Metropolis algorithm is used to iteratively update the latent space variable $\mathbf{z}$. An extension of this model was subsequently developed, which incorporates a loudness gain for robust speech modeling and adopts a noise model based on alpha-stable distributions [23], [36]. The Monte Carlo expectation-maximization algorithms were used for estimating the model parameters.

To the best of our knowledge, the idea of incorporating the VAE concept into the multichannel framework was first introduced in a preprint article [41] and later published as a journal paper [21]. Unlike the above VAE-NMF methods, this method, namely the MVAE method, uses a CVAE with a fully convolutional architecture to model the entire spectrogram of an utterance of each source. While the original MVAE method was designed to deal with determined anechoic mixtures only, its modified versions have subsequently been proposed to handle underdetermined scenarios [34] and highly reverberant conditions [39]. Like the original version, these two versions use gradient descent (backpropagation) to update the source model parameters. Extensions of the VAE-NMF methods to multichannel inputs were later developed [35], [37], [42] for application to multichannel speech enhancement tasks. In these methods, the Markov chain Monte Carlo (MCMC) methods [43], such as Gibbs sampling

and the Metropolis algorithm, are used to iteratively update the latent space variable as with the original VAE-NMF methods.

Although these methods have been shown to perform impressively compared with conventional NMF-based methods, the use of sampling and backpropagation to update latent space variables can be computationally expensive. To reduce the computational cost, we previously proposed to exploit the pretrained encoder of a CVAE as an approximate posterior estimator to infer the latent space variable $\mathbf{z}$ in [27]. With the same motivation, a fast algorithm for estimating the parameters of the VAE-NMF model was later derived based on the Bayesian inference in [38] for single-channel speech enhancement.

IV. FMVAE ALGORITHM

A. IDEA

In this section, we describe the idea of the proposed fast optimization algorithm for the MVAE method. Since the process of updating the parameters of the CVAE source model is more computationally costly than that of updating the other parameters, our main focus is on how to accelerate this process. When $\mathcal{W}$ is fixed, each element of $\boldsymbol{S}_j$ will be fixed at $s_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\,\mathbf{x}(f,n)$. Now, since the terms that depend on $\mathbf{z}_j$ and $\mathbf{c}_j$ in (16) are given as

$$\begin{aligned}&\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)+\log p(\mathbf{z}_j)+\log p(\mathbf{c}_j)\\ &\qquad\stackrel{c}{=}\log p_\theta(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j),\end{aligned}\tag{22}$$

we would like to find $\mathbf{z}_j$ and $\mathbf{c}_j$ that maximize the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ after updating $\mathcal{W}$. This posterior can be factorized as $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j) = p(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)p(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$. Here, we notice that the first factor, $p(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$, resembles the encoder (or inference) distribution in the CVAE in (11), with the difference being that it is also conditioned on the scale parameter g_j. Since the total energy of each training utterance is assumed to be normalized to 1 in the CVAE training as mentioned earlier, g_j can be thought of as a parameter that plays the role of normalizing the total energy of an unnormalized input $\boldsymbol{S}_j$ to 1 at test time so that the scale of the encoder input is ensured to be consistent with the training utterances. Specifically, the encoder distribution that allows for unnormalized inputs is implicitly assumed to be given as the following expression:

$$\begin{aligned}q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c},g)&=\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\boldsymbol{S}/g,\mathbf{c}),\mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{S}/g,\mathbf{c})))\\ &=\prod_k\mathcal{N}(z(k)|\mu_\phi(k;\boldsymbol{S}/g,\mathbf{c}),\sigma^2_\phi(k;\boldsymbol{S}/g,\mathbf{c})),\end{aligned}\tag{23}$$

which reduces to (11) when $g = 1$. Thus, we can use the trained encoder $q_\phi(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$ as an approximation of the first factor of the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$. This means that if we could obtain the true distribution $p(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ or its approximate distribution $r(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$, we would be able to find an approximation of the maximum point of the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ by finding the maximum point of the corresponding approximate distribution.

In this section, we review the concept of an auxiliary classifier VAE (ACVAE), present how this concept can be used to obtain $r(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$, and introduce the details of the proposed optimization algorithm.

B. AUXILIARY CLASSIFIER VAE

An auxiliary classifier VAE (ACVAE) [26] is a CVAE variant, which incorporates an information-theoretic regularization [44] that assists in making the decoder outputs as correlated as possible with the class variable $\mathbf{c}$ by maximizing the mutual information between $\mathbf{c}$ and an output $\boldsymbol{S} \sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$ from the decoder, conditioned on $\mathbf{z}$. The mutual information is expressed as

$$I(\mathbf{c},\boldsymbol{S}|\mathbf{z})=\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}),\mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log p(\mathbf{c}'|\boldsymbol{S})]+H(\mathbf{c}),\tag{24}$$

where $p_D(\mathbf{c})$ is the empirical distribution of $\mathbf{c}$ in the training dataset, and $H(\mathbf{c})$ represents the entropy of $\mathbf{c}$, which can be considered as a constant term. Although it is difficult to optimize $I(\mathbf{c},\boldsymbol{S}|\mathbf{z})$ directly since it requires access to the posterior $p(\mathbf{c}|\boldsymbol{S})$, we can derive a variational lower bound of the first term of $I(\mathbf{c},\boldsymbol{S}|\mathbf{z})$ by using a variational distribution $r(\mathbf{c}|\boldsymbol{S})$ to approximate $p(\mathbf{c}|\boldsymbol{S})$:

$$\begin{aligned}&\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}),\mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log p(\mathbf{c}'|\boldsymbol{S})]\\ &\quad=\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}),\mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}\Big[\log\frac{r(\mathbf{c}'|\boldsymbol{S})p(\mathbf{c}'|\boldsymbol{S})}{r(\mathbf{c}'|\boldsymbol{S})}\Big]\\ &\quad=\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}\big[\mathrm{KL}[p(\mathbf{c}'|\boldsymbol{S})||r(\mathbf{c}'|\boldsymbol{S})]\\ &\qquad+\mathbb{E}_{\mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log r(\mathbf{c}'|\boldsymbol{S})]\big]\\ &\quad\geq\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}),\mathbf{c}'\sim p(\mathbf{c}|\ VecS)}[\log r(\mathbf{c}'|\boldsymbol{S})]\\ &\quad=\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}[\log r(\mathbf{c}|\boldsymbol{S})],\end{aligned}\tag{25}$$

where the equality holds if and only if $r(\mathbf{c}|\boldsymbol{S}) = p(\mathbf{c}|\boldsymbol{S})$. This technique of lower bounding mutual information is known as variational information maximization [45]. The last line of (25) follows the lemma presented in [44]. Therefore, we can indirectly maximize $I(\mathbf{c},\boldsymbol{S}|\mathbf{z})$ by increasing the lower bound with respect to $p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$ and $r(\mathbf{c}|\boldsymbol{S})$. One way to achieve this involves expressing the variational distribution $r(\mathbf{c}|\boldsymbol{S})$ as a neural network and training it along with $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$ and $p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})$. Specifically, $r(\mathbf{c}|\boldsymbol{S})$ can be expressed as a multinomial distribution

$$r_\psi(\mathbf{c}|\boldsymbol{S}) = \mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}_\psi(\boldsymbol{S})).\tag{26}$$

Here, $\mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}) \propto \prod_i \rho_i^{c_i}$ denotes a multinomial distribution, where $\mathbf{c} = [c_1,\ldots,c_I]^{\mathsf{T}}$ and $\boldsymbol{\rho} = [\rho_1,\ldots,\rho_I]^{\mathsf{T}}$. $\boldsymbol{\rho}_\psi(\boldsymbol{S})$ denotes a neural network that takes $\boldsymbol{S}$ as an input and produces a probability vector consisting of C elements. (26) is called an auxiliary classifier.

Therefore, the regularization term that we would like to maximize over the training samples with respect to ϕ, θ, and

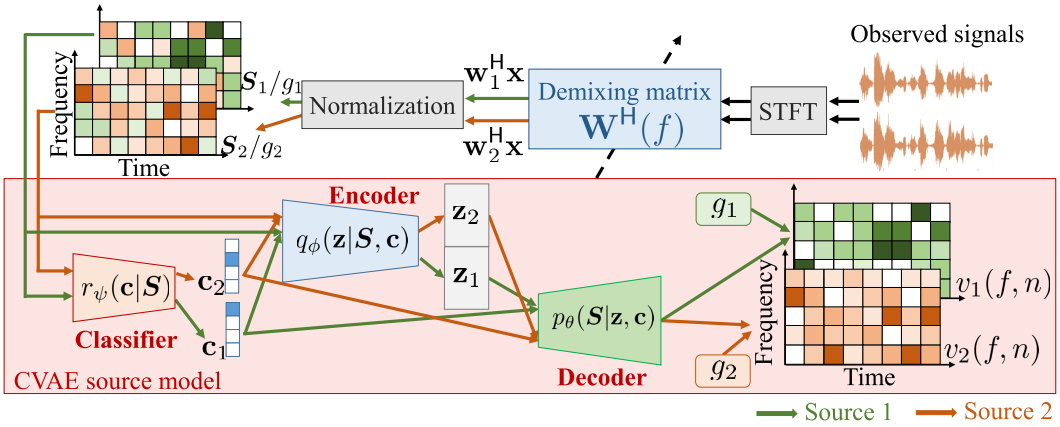

FIGURE 3. Flowchart of fMVAE for $I = 2$ case.

ψ becomes

$$\mathcal{L}(\phi,\theta,\psi) = \mathbb{E}_{(\boldsymbol{S},\mathbf{c})\sim p_D(\boldsymbol{S},\mathbf{c}),q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})}[\mathbb{E}_{\mathbf{c}\sim p_D(\mathbf{c}),\boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}[\log r_\psi(\mathbf{c}|\boldsymbol{S})]], \quad (27)$$

where $r_\psi(\mathbf{c}|\boldsymbol{S})$ must satisfy the sum-to-one constraint. With the regularization term (27), the auxiliary classifier is trained using only the reconstructed spectrograms. Since we can also use the spectrograms of real speech to train the auxiliary classifier, we can further use the cross-entropy

$$\mathcal{I}(\psi) = \mathbb{E}_{(\boldsymbol{S},\boldsymbol{c})\sim p_D(\boldsymbol{S},\boldsymbol{c})}[\log r_\psi(\boldsymbol{c}|\boldsymbol{S})] \quad (28)$$

as the training criterion. The entire training criterion is thus given by

$$\mathcal{J}(\phi,\theta) - \lambda_{\mathcal{L}}\mathcal{L}(\phi,\theta,\psi) - \lambda_{\mathcal{I}}\mathcal{I}(\psi), \quad (29)$$

where $\lambda_{\mathcal{L}} \geq 0$ and $\lambda_{\mathcal{I}} \geq 0$ are the parameters weighing the importance of the regularization terms. Figure 2 shows an illustration of ACVAE.

C. FAST ALGORITHM

As mentioned above, the auxiliary classifier distribution $r_\psi(\mathbf{c}|\boldsymbol{S})$ trained using $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$ is expected to be a good approximation of the conditional distribution $p(\mathbf{c}|\boldsymbol{S})$. Now, in the same way that we considered the encoder that flexibly allows for an unnormalized input, here we also consider an auxiliary classifier $r_\psi(\mathbf{c}|\boldsymbol{S}, g)$ that incorporates the global scale parameter g such that

$$r_\psi(\mathbf{c}|\boldsymbol{S}, g) = \mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}_\psi(\boldsymbol{S}/g)). \quad (30)$$

Using the trained auxiliary classifier and encoder, we can obtain an approximation $p(\mathbf{z}_j, \mathbf{c}_j|\boldsymbol{S}_j, g_j) \approx r_\psi(\mathbf{c}_j|\boldsymbol{S}_j, g_j)\, q_\phi(\mathbf{z}_j|\boldsymbol{S}_j, \mathbf{c}_j, g_j)$. Since the maximum points of $r_\psi(\mathbf{c}_j|\boldsymbol{S}_j, g_j)$ and $q_\phi(\mathbf{z}_j|\boldsymbol{S}_j, \mathbf{c}_j, g_j)$ can be found immediately, we can use these approximate distributions to find an approximate solution to $(\mathbf{z}_j, \mathbf{c}_j) = \mathrm{argmax}_{\mathbf{z}_j,\mathbf{c}_j} p(\mathbf{z}_j, \mathbf{c}_j|\boldsymbol{S}_j, g_j)$ instead of the gradient descent update for increasing $\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j, \mathbf{c}_j, g_j) + \log p(\mathbf{z}_j) + \log p(\mathbf{c}_j)$. Figure 3 shows the flowchart of the proposed algorithm for the $I = 2$ case. The algorithm is summarized in *Algorithm 2*. The main difference between the new algorithm from the original version is that the optimal $\mathbf{z}_j$ and $\mathbf{c}_j$ are estimated using the forward propagations of the two pretrained networks instead of using gradient descent updates. Specifically, $\mathbf{z}_j$ is given as the mean of the encoder distribution $\boldsymbol{\mu}_\phi(\boldsymbol{S}_j/g_j, \mathbf{c}_j)$. There are two possible ways to update the class variable $\mathbf{c}_j$. One is to directly use the probability vector produced by the auxiliary classifier network

$$\mathbf{c}_j \leftarrow \boldsymbol{\rho}_\psi(\boldsymbol{S}_j/g_j). \quad (31)$$

We hereafter refer to the proposed algorithm using this update rule as *fMVAE_c*. The other is to use the one-hot vector closest to the output of the auxiliary classifier

$$[\mathbf{c}_j]_k \leftarrow \begin{cases} 1, & (k = \hat{k}), \\ 0, & (k \neq \hat{k}), \end{cases} \quad (32)$$

$$\hat{k} = \underset{k}{\mathrm{argmax}}\ [\boldsymbol{\rho}_\psi(\boldsymbol{S}_j/g_j)]_k, \quad (33)$$

where $[\cdot]_k$ is used to denote the kth element of a vector. We hereafter refer to the algorithm using this update rule as *fMVAE_o*. Here, the subscripts are the first letters of "continuous" and "one-hot", respectively. $r_\psi(\mathbf{c}_j|\boldsymbol{S}_j, g_j)$ can be seen as a speaker recognizer trained with explicit supervision. Hence, the proposed algorithm is expected to perform better than the original version in terms of speaker identification accuracy. However, one downside would be that it does not guarantee a non-decrease in the objective function because of the approximation $p(\mathbf{z}_j, \mathbf{c}_j|\boldsymbol{S}_j, g_j) \approx r_\psi(\mathbf{c}_j|\boldsymbol{S}_j, g_j) q_\phi(\mathbf{z}_j|\boldsymbol{S}_j, \mathbf{c}_j, g_j)$. How this actually affects source separation performance will be discussed later.

D. PRIOR-WEIGHTED INFERENCE

The encoder network is trained so that $q_\phi(\mathbf{z}|\boldsymbol{S}, \mathbf{c})$ becomes as close as possible to $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. However, through preliminary experiments, we found that at test time the trained encoder occasionally produced outliers that significantly deviated from the assumed distribution $\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. This may be because the encoder did not generalize very well

Algorithm 2 fMVAE Algorithm

Require: Network parameter θ, ϕ, ψ trained using (29), observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$, weight parameter α

randomly initialize $\mathcal{W}$, Ψ
optional: update $\mathcal{W}$ using a BSS method
for $\ell = 1$ to $\mathscr{L}$ **do**
  for each source j of J **do**
    $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
    (updating source model paremeters)
    initialize g_j using (19)
    normalization: $\bar{S}_j = \{y_j(f,n)/g_j\}_{f,n}$
    update $\mathbf{c}_j$ using (31) or (32)
    update $\mathbf{z}_j$ using (36)
    compute $\sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    update g_j using (19)
    compute $v_j(f,n) = g_j \cdot \sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    (updating demixing matrices)
    update $\mathbf{w}_j(f)$ by IP method with (17), (18)
  end for
end for

due to the limited amount of training data or the mismatch between the training and test conditions. Since the decoder network was trained under the assumption that its input follows $\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I})$, these outliers tended to negatively affect the resulting decoder outputs and eventually the estimate of $\mathbf{W}$. One heuristic way to address this problem would be to reapply the prior distribution $p(\mathbf{z})$ during inference. In the following, we omit the source index j in this subsection for simplicity of notation.

As a way of reapplying the prior, we adopt the concept of product-of-experts (PoE) [28] and define $\hat{\mathbf{z}}$ as

$$\begin{aligned}\hat{\mathbf{z}} &= \operatorname*{argmax}_{\mathbf{z}} p(\mathbf{z}|\boldsymbol{S},\mathbf{c},g)p(\mathbf{z})^{\alpha} \\ &\approx \operatorname*{argmax}_{\mathbf{z}} q_{\phi}(\mathbf{z}|\boldsymbol{S},\mathbf{c},g)p(\mathbf{z})^{\alpha} \\ &= \operatorname*{argmax}_{\mathbf{z}} \log q_{\phi}(\mathbf{z}|\boldsymbol{S},\mathbf{c},g) + \alpha \log p(\mathbf{z}), \end{aligned} \tag{34}$$

where α weighs the importance of the prior in the inference. Since both $q_{\phi}(\mathbf{z}|\boldsymbol{S},\mathbf{c},g)$ and $p(\mathbf{z})$ are multivariate Gaussian distributions, (34) can be expressed as

$$\begin{aligned}&\log q_{\phi}(\mathbf{z}|\boldsymbol{S},\mathbf{c},g) + \alpha \log p(\mathbf{z}) \\ &=^{c} -\frac{1}{2}\big(\mathbf{z}-\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g,\mathbf{c})\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\phi}^{-1}\big(\mathbf{z}-\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g,\mathbf{c})\big) - \frac{\alpha}{2}\mathbf{z}^{\mathsf{T}}\mathbf{z} \\ &=^{c} -\frac{\boldsymbol{\Sigma}_{\phi}^{-1}+\alpha\mathbf{I}}{2}(\mathbf{z}-\boldsymbol{\mu})^{\mathsf{T}}(\mathbf{z}-\boldsymbol{\mu}), \end{aligned} \tag{35}$$

where $\boldsymbol{\Sigma}_{\phi} = \mathrm{diag}(\boldsymbol{\sigma}_{\phi}^2(\boldsymbol{S}/g,\mathbf{c}))$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}_{\phi}^{-1}(\boldsymbol{\Sigma}_{\phi}^{-1} + \alpha\mathbf{I})^{-1}\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g,\mathbf{c})$. Therefore, the update rule for $\mathbf{z}$ can be easily derived as

$$\mathbf{z} \leftarrow \boldsymbol{\Sigma}_{\phi}^{-1}(\boldsymbol{\Sigma}_{\phi}^{-1}+\alpha\mathbf{I})^{-1}\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g,\mathbf{c}). \tag{36}$$

Note that (36) reduces to the mean of the encoder distribution when $\alpha = 0$.

E. POTENTIAL ADVANTAGE OF CVAE OVER REGULAR VAE IN TERMS OF SOURCE MODELING

Although the MVAE method uses a CVAE for source spectral modeling, one can also think of using a regular (unconditional) VAE, as in the VAE-NMF framework. In this case, all the factors of variations in speech spectra, including the speaker identity factor, will be encoded into the latent variables. However, this can lead to an overparametrized representation since even though the speaker identity factor should be considered time-invariant (unlike phoneme- and F_0-related factors), the latent variables are allowed to vary over time. Hence, when estimating the latent variable sequence of each source in a given mixture, we would want to separate out only the speaker identity factor from the latent variable sequence and force it to be time-invariant so as not to allow it to change during the utterance. This is the motivation behind the idea of using a CVAE instead of a regular VAE. A quantitative comparison between these choices is provided in Subsection V-E.

V. EXPERIMENTAL EVALUATIONS

To evaluate the effectiveness of the proposed method, we conducted several multi-speaker source separation experiments in which we considered speaker-dependent and speaker-independent separation tasks. Specifically, the speaker-dependent and speaker-independent conditions indicate whether the test speaker is seen in the training dataset. It should be noted that even in the speaker-dependent condition, the training and test sets are disjoint at the sentence level. In this section, we first provide the details of the baseline algorithms in Subsection V-A and the network architectures used in the baseline and proposed methods in Subsection V-B. We then show how the dataset was created and present the experimental results obtained under the speaker-dependent condition in Subsections V-C – V-F. In Subsection V-G, we describe the large-scale dataset designed for the speaker-independent task and show the experimental results.

A. BASELINE METHODS FOR COMPARISON

We chose ILRMA [6], IDLMA [46], and the original MVAE method [2] [21] as the baseline methods for comparison. We tested several different versions of the proposed and baseline methods. We use the terms "supervised/unsupervised" and "informed/uninformed" to properly categorize each version of the methods. The terms "supervised" and "unsupervised" indicate whether a method requires training examples of source signals prior to source separation, while the terms "informed" and "uninformed" indicate whether a method is informed about which sources are present in a test mixture signal. Categorization of each version is summarized in Table 1.

We set the basis number $T_j = 10$ for u.u.ILRMA and randomly initialized the basis spectra and activation matrix.

[2]Code: https://github.com/lili-0805/MVAE

TABLE 1. Methods for comparison.

Category	Method	Notation	Initialization
unsupervised uninformed	ILRMA	Baseline1: u.u.ILRMA	random
supervised uninformed	ILRMA	Baseline2: s.u.ILRMA	random
	MVAE	Baseline3: s.u.MVAE	random/IVA/u.u.ILRMA
	fMVAE_o	Proposed1: s.u.fMVAE_o	random/IVA/u.u.ILRMA
	fMVAE_c	Proposed2: s.u.fMVAE_c	random/IVA/u.u.ILRMA
supervised informed	ILRMA	Baseline4: s.i.ILRMA	random
	IDLMA	Baseline5: s.i.IDLMA	random
	MVAE	Baseline6: s.i.MVAE	random/IVA/u.u.ILRMA
	fMVAE	Proposed3: s.i.fMVAE	random/IVA/u.u.ILRMA

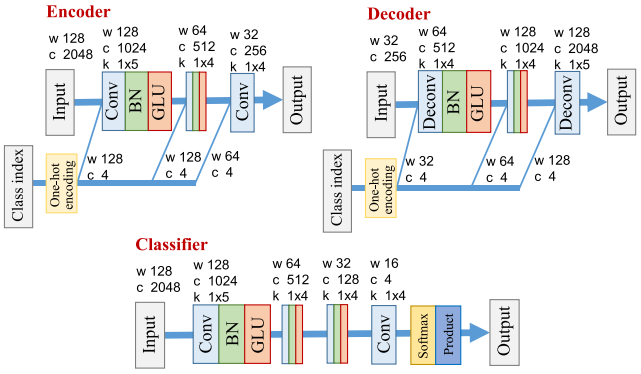

FIGURE 4. Network architectures of the encoder and decoder used for MVAE and fMVAE and the classifier used for fMVAE. The inputs and outputs are one-dimensional data, where the frequency dimension of the spectrograms is regarded as the channel dimension. The 'w', 'c', and 'k' denote the width, channel number, and kernel size, respectively. Conv and Deconv denote one-dimensional convolution and deconvolution; BN and GLU stand for batch normalization and gated linear unit.

For supervised ILRMA, basis spectra with $T = 10$ were pre-trained for each speaker in the training dataset using the NMF algorithm. They were then concatenated and used as a unified model to represent all the sources in s.u.ILRMA, whereas the basis spectra corresponding to the specific speakers present in a mixture signal were provided to the method in s.i.ILRMA.

B. NETWORK ARCHITECTURES

Figure 4 depicts the details of the network architectures employed in the MVAE and fMVAE methods. We used the same network architectures to train the CVAE and ACVAE. All the networks were designed to be fully convolutional to handle input spectrograms of signals with arbitrary lengths. We used one-dimensional gated convolutional neural networks (CNNs) [47] to model spectrograms, which allows the networks to capture time dependencies in spectral sequences. Gated CNNs were initially introduced to model word sequences for language modeling and shown to outperform long short-term memory (LSTM) language models trained in a similar setting. The effectiveness of employing a gated CNN to model a spectrogram has already been confirmed [48], [49]. By using $\mathbb{O}_{l-1}$ to denote the output of the $(l-1)$th layer, the output of the lth layer $\mathbb{O}_l$ of a gated CNN

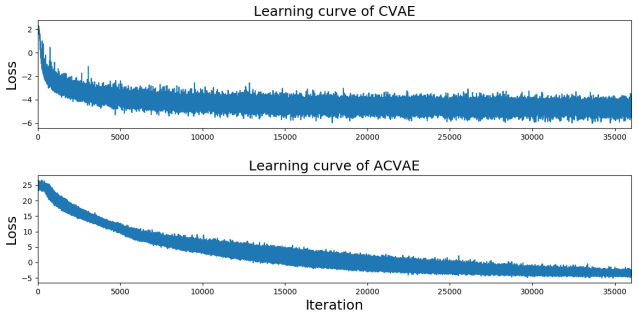

FIGURE 5. Learning curves of CVAE and ACVAE source models.

can be written as

$$\mathbb{O}_l = (\mathbb{O}_{l-1} * \mathbb{W}_l^{\mathrm{f}} + \mathbb{b}_l^{\mathrm{f}}) \otimes \sigma(\mathbb{O}_{l-1} * \mathbb{W}_l^{\mathrm{g}} + \mathbb{B}_l^{\mathrm{g}}), \quad (37)$$

where $\mathbb{W}_l^{\mathrm{f}}$, $\mathbb{W}_l^{\mathrm{g}}$, $\mathbb{B}_l^{\mathrm{f}}$, and $\mathbb{B}_l^{\mathrm{g}}$ are weight and bias parameters of the lth layer, $\otimes$ denotes element-wise multiplication, and σ is the sigmoid function. The main difference between a gated CNN and a regular CNN layer is that a gated linear unit (GLU), namely the second term of (37), is used as a nonlinear activation function. Like LSTMs, GLUs have data-driven gates, which control the information passed on in the hierarchy. At each gated CNN layer in the encoder and decoder, a broadcast version of c is appended along the channel dimension to the output of the previous layer. Adam [50] was used to train the networks. Note that *Algorithm 1* and *Algorithm 2* correspond to s.u.MVAE and s.u.fMVAE_o/s.u.fMVAE_c, respectively. For s.i.MVAE and s.i.fMVAE, the correct class label $\mathbf{c}_j$ is given and fixed during the update. Figure 5 shows the learning curves of the CVAE and ACVAE training processes. The curves demonstrate that the networks were trained stably with fast convergence.

For s.i.IDLMA, we used a fully connected neural network with four hidden layers. Each layer had 1024 units, and a rectified linear unit was used for the output of each layer, which was the same as the network architecture used in [46]. We implemented the training settings described in [46], namely using the Gaussian-IDLMA loss function and concatenation of the current, preceding, and succeeding frames to capture the temporal dependency, data augmentation, and regularization. The only difference was the optimization algorithm, where we used Adam to train the network for 700 epochs instead of Adadelta [51] for 200 epochs. More training details are available in [46].

C. DATASET FOR SPEAKER-DEPENDENT SEPARATION

As in the original MVAE paper [21], we used speech utterances of two male speakers (SM1, SM2) and two female speakers (SF1, SF2) excerpted from the Voice Conversion Challenge (VCC) 2018 dataset [52] for the speaker-dependent source separation experiment. The audio files for each speaker were about seve minutes long and manually segmented into 116 short sentences, where 81 and

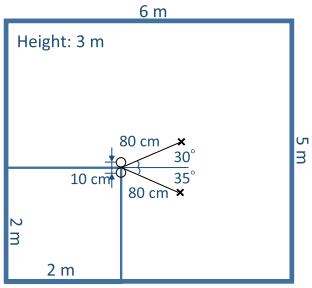

FIGURE 6. Configuration of room, where ∘ and × represent the positions of microphones and sources, respectively.

35 sentences (about five and two minutes long, respectively) served as training and test sets, respectively.

We used two-channel mixture signals of two sources as the test data, which were synthesized using simulated room impulse responses (RIRs) generated using the image method [53] and real RIRs measured in an anechoic room (ANE) and an echo room (E2A). Figure 6 shows the configuration of the room used for simulating RIRs. To meet the instantaneous mixing model assumption, the reverberation times (RT_{60}) [54] of the simulated RIRs were set at 78 and 351 ms, which were controlled by setting the reflection coefficient of the walls at 0.20 and 0.80, respectively. For the measured RIRs, we used the data included in the RWCP Sound Scene Database in Real Acoustic Environments [55]. The RT_{60} of ANE and E2A were 173 and 225 ms, respectively. The test data included 4 pairs of speakers, i.e., SF1+SF2, SF1+SM1, SM1+SM2, and SF2+SM2. For each speaker pair, we generated ten mixture signals. Hence, there were a total of 40 test signals for each reverberation condition, each of which was about four to seven seconds long. All the speech signals were resampled at 16 kHz.

D. EXPERIMENTAL ANALYSIS OF WINDOW LENGTH, INITIALIZATION, AND WEIGHT PARAMETER α

In this subsection, we compare the separation performance across different STFT window lengths, different initialization methods for the MVAE and fMVAE algorithms, and different α settings.

Since all the methods are based on the instantaneous linear mixture model, the STFT window length may affect the separation performance of each of them, especially under reverberant conditions. We computed the STFT using a Hamming window with a length of {32, 64, 128, 256} ms, and by shifting half of the length for each frame. In this experiment, all the MVAE and fMVAE methods were initialized by running u.u.ILRMA for 30 iterations. The MVAE or fMVAE algorithm was then run for 30 iterations, where Adam was used to update $\mathbf{z}_j$ and $\mathbf{c}_j$ in the MVAE methods with a step size set of 0.01. We used $\alpha = 0$ for fMVAE in this experiment. Table 2 shows the SDR scores obtained with each method. From these results, the optimal window length that gave the best overall performance was 128 ms for the current dataset.

TABLE 2. Average SDR [dB] obtained with various STFT settings. The bold font shows the best scores.

Method	Window length [ms]			
	32	64	128	256
s.u.MVAE	10.91	13.38	**14.01**	12.27
s.i.MVAE	10.84	13.41	**13.76**	12.47
s.u.fMVAE_o	11.63	12.11	**14.67**	13.85
s.u.fMVAE_c	4.31	10.36	**13.51**	13.26
s.i.fMVAE	11.57	12.25	**14.76**	14.13

TABLE 3. Average SDR [dB] obtained by MVAE and fMVAE methods adopting different initialization approaches. The bold font shows the best scores.

Method	Initialization		
	random	IVA	ILRMA
s.u.MVAE	**17.03**	12.58	14.01
s.i.MVAE	**16.58**	12.45	13.76
s.u.fMVAE_o	14.26	13.67	**14.67**
s.u.fMVAE_c	**13.78**	12.62	13.51
s.i.fMVAE	**14.93**	13.82	14.76

Therefore, we conducted all the following experiments using a window length of 128 ms.

To confirm the impact of the initialization for the MVAE and fMVAE methods on the source separation performance, we compared the algorithms using the following three initialization methods: 1) random initialization with the demixing matrices initialized at identity matrices; 2) IVA; and 3) u.u.ILRMA. To keep the number of updates of the demixing matrices constant, each algorithm was run for 60 iterations for the random initialization case and 30 iterations after an initialization algorithm was run for 30 iterations for the other cases. Table 3 shows the SDR scores over the 160 test samples. From these results, we found that the methods adopting ILRMA for initialization achieved better performance than those using IVA for initialization. One possible reason could be that block permutation had occurred in IVA. It is worth noting that the MVAE methods with random initialization obtained more than 3 dB higher SDR improvements than when using IVA and ILRMA for initialization. Meanwhile, though random initialization slightly outperformed ILRMA in s.u.fMVAE_c and s.i.fMVAE, there were no noticeable differences. Therefore, we adopted random initialization in the following experiments.

Finally, we investigated how much the performance depends on the weight parameter α in the prior-weighted inference. We set α at {0, 1, 10, 50, 100, 200, 300, mean}, where "mean" indicates the data-dependent setting

$$\alpha = \frac{1}{K}\sum_{k}^{K}\sigma_{\phi}^{2}(k; \boldsymbol{S}, \mathbf{c}). \tag{38}$$

Figure 7 shows the average SDR scores over 160 test signals. We found that the effectiveness of the prior distribution $p(\mathbf{z})$ in improving the source separation performance was modest in the speaker-dependent case and that the SDRs started to decrease at $\alpha > 10$, which indicates that a smaller value

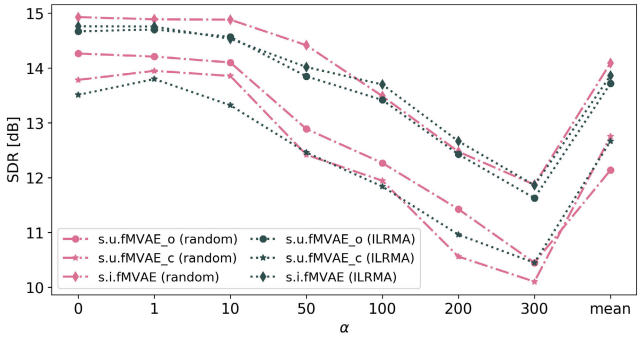

FIGURE 7. Average SDR achieved with various α in a speaker-dependent condition.

TABLE 4. Average SDR, SIR, SAR, PESQ, and STOI scores achieved by MVAE with CVAE and VAE for source modeling. The bold font indicates the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
s.u.MVAE(VAE)	15.35	20.30	17.91	**2.72**	0.8495
s.u.MVAE(CVAE)	**17.03**	**23.75**	**18.61**	2.24	**0.8717**

leads to better performance for the speaker-dependent case. Moreover, the curve of fMVAE_o was entirely above the curve of fMVAE_c without regard for the choice of the initialization methods, which indicates that fMVAE_o is more effective in speaker-dependent scenarios.

E. SOURCE SEPARATION PERFORMANCE

In addition to SDRs, we used signal-to-interference ratios (SIRs) and signal-to-artifact ratios (SARs) [56] to evaluate the source separation performance. Perceptual evaluations of speech quality (PESQ)[3] [57] and short-time objective intelligibility (STOI)[4] [58] were also conducted to ascertain the speech quality and intelligibility. All the criteria were calculated using a dry source as the reference signal.

We first confirmed the effectiveness of conditional modeling by comparing the performance obtained with the CVAE source model and its unconditional counterpart under the MVAE framework. Table 4 shows SDR, SIR, SAR, PESQ, and STOI scores. As can be seen from the results, the CVAE source model obtained a 1.7-dB higher SDR than a source model based on a regular VAE.

Table 5 shows scores obtained by each method with the optimal parameter setting. By comparing supervised methods to the blind method (u.u.ILRMA), we confirmed that an appropriately pretrained source model could lead to considerably improved source separation performance. The MVAE methods achieved the best scores in both the uninformed and informed categories, which significantly outperformed the other methods. The fMVAE method yielded an average SDR score that was 2.8 dB lower than the original MVAE method, but about 0.75 dB higher than the other baseline methods.

[3]Code: https://github.com/vBaiCai/python-pesq
[4]Code: https://github.com/mpariente/pystoi

TABLE 5. Average SDR, SIR, SAR, PESQ, and STOI scores achieved by each method with the optimal parameter setting. The bold font indicates the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
u.u.ILRMA	12.36	17.77	15.29	1.83	0.8345
s.u.ILRMA	13.50	19.01	16.60	1.92	0.8367
s.u.MVAE	**17.03**	**23.75**	**18.61**	2.24	**0.8717**
s.u.fMVAE_o	14.26	19.89	16.71	2.07	0.8454
s.u.fMVAE_c	13.95	19.54	16.33	2.66	0.8452
s.i.ILRMA	13.30	18.60	17.02	1.91	0.8355
s.i.IDLMA	14.15	21.11	15.59	1.77	0.8692
s.i.MVAE	16.58	22.87	18.40	**2.84**	0.8641
s.i.fMVAE	14.93	21.00	16.98	2.73	0.8548

TABLE 6. Computational times of MVAE and fMVAE methods with random initialization.

Processor	Method	rumtime/iteration [sec]	total [sec]
GPU	s.u.MVAE	2.8147	172.5241
	s.u.fMVAE_o	**0.0367**	**5.5661**
	s.u.fMVAE_c	**0.0365**	**5.5372**
CPU	s.u.fMVAE_o	0.0979	8.6823
	s.u.fMVAE_c	0.0969	8.7434

F. COMPUTATIONAL TIME

The average computational times of the MVAE and fMVAE methods with random initialization are summarized in Table 6. All the programs were run using an Intel (R) Core i7-7800X CPU@3.50 GHz and a TITAN V GPU with 12-GB memory. Here, "runtime/iteration" means the computational time required to update the parameters once using the MVAE or fMVAE algorithm. The "total" time indicates the time taken by the entire process, including the time for constructing the system (e.g., loading the pretrained networks to a GPU), updating parameters, and performing the separation. Through the comparison of the runtime at each iteration, we found that the fMVAE algorithm was about 70 times faster than the MVAE algorithm. Moreover, fMVAE was found to reduce the computational time by more than 90% even when using a CPU. These results indicate a tradeoff between the source separation performance and computational time: the MVAE method provides better separation performance with high computational cost, whereas fMVAE significantly reduces computational cost but with performance degradation.

G. SPEAKER-INDEPENDENT SEPARATION

In practical applications, the speakers in a given mixture signal are not always included in the training dataset. In this subsection, we show the performance of the MVAE and fMVAE methods in speaker-independent tasks and compare them with u.u.ILRMA, which requires no prior information about the speakers.

We created datasets using utterances from the Wall Street Journal (WSJ0) corpus [59]. All the utterances in WSJ0 folder `si_tr_s` (around 25 hours) were used as the training set, which consists of 101 speakers in total. If there is a large number of utterances of a sufficiently wide variety of speakers

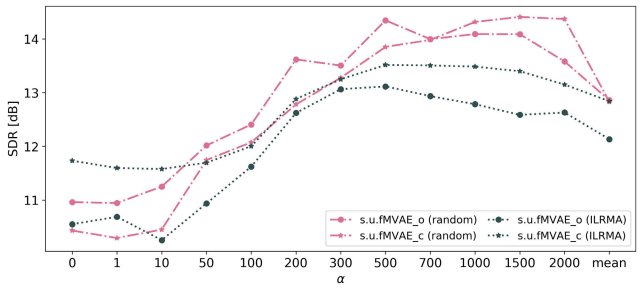

FIGURE 8. Average SDR over 200 test signals achieved with various α.

TABLE 7. Average SDR, SIR, SAR, PESQ, and STOI scores obtained with uninformed methods. The bold font shows the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
u.u.ILRMA	13.76	19.94	17.09	**3.05**	0.8727
s.u.MVAE	**17.58**	**25.13**	**19.26**	2.65	**0.8934**
s.u.fMVAE_o	14.35	21.06	17.25	3.04	0.8746
s.u.fMVAE_c	14.41	21.21	17.35	3.04	0.8776

in the training dataset, the trained model is expected to have an ability to express spectrograms of unseen speakers. When a test mixture contains unseen speakers, (31) can be interpreted as how similar speaker j is to the speakers in the training set, whereas (32) indicates the speaker in the training set most similar to speaker j. A test set was created by randomly mixing two different speakers selected from the WSJ0 folders `si_dt_05` and `si_et_05`, where the number of speakers was 18. We generated test data using simulated RIRs with $RT_{60} = 78$ ms and $RT_{60} = 351$ ms, where 100 mixture signals were generated under each reverberation condition. The average SDRs of the datasets were about 0.60 dB and -0.78 dB, respectively. Other experimental conditions and network architectures were the same as those described in Subsection V-C.

As in the speaker-dependent case, we first investigated the dependence of the separation performance on the α setting. Figure 8 shows the average SDR scores over the entire test dataset achieved with various α settings. Since the scores obtained with $\alpha = 200$ and $\alpha = 300$ increased continuously, we additionally evaluated the performance obtained when $\alpha = \{500, 700, 1000, 1500, 2000\}$. The optimal α settings were 500 for s.u.fMVAE_o and 2000 for s.u.fMVAE_c, respectively. This was considerably different from the speaker-dependent case, where a smaller α performed better. From these results, we can assume that the proposed prior-weighted update rule was more effective under open-set conditions than under closed-set conditions.

Table 7 summarizes the average SDR, SIR, SAR, PESQ, and STOI scores obtained with each method with random initialization. The results demonstrate the ability of the MVAE and fMVAE methods to handle speaker-independent scenarios with an increasing variety and amount of training data. Both the MVAE and fMVAE methods were superior to u.u.ILRMA, where s.u.MVAE achieved an improvement of more than 3.5 dB over u.u.ILRMA. As with the speaker-dependent case, the fMVAE methods provided less improvement than the MVAE method.

VI. CONCLUSION

This paper proposed a novel optimization algorithm for the MVAE method, which is called FastMVAE (or fMVAE). The proposed method exploits an auxiliary classifier VAE instead of a regular CVAE to learn the generative distribution of source signals and employs the trained auxiliary classifier and encoder for inference. We newly introduced a prior-weighted update rule for the latent variables of each CVAE source model and different update rules for the class label of each source. We conducted experiments to investigate the optimal window length, initialization, and weight parameter and performed speaker-dependent and speaker-independent source separation experiments to confirm the effectiveness of the proposed method. Experimental results revealed that fMVAE can significantly reduce computational time by more than 90% compared with the original MVAE method; the MVAE and fMVAE methods outperformed conventional methods under speaker-dependent conditions; and the MVAE and fMVAE methods can handle a speaker-independent scenario by using a large set of training data.

ACKNOWLEDGMENT

This article was presented in part at the ICASSP 2019 as a conference paper [27].

REFERENCES

[1] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ICA to multivariate components," in *Proc. Integr. Comput.-Aided Eng.*, 2006, pp. 165–172.
[2] A. Hiroe, "Solution of permutation problem in frequency domain ICA using multivariate probability density functions," in *Proc. Integr. Comput.-Aided Eng.*, 2006, pp. 601–608.
[3] A. Ozerov and C. Fevotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 550–563, Mar. 2010.
[4] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 5, pp. 971–982, May 2013.
[5] H. Kameoka, T. Yoshioka, M. Hamamura, J. Le Roux, and K. Kashino, "Statistical model of speech signals based on composite autoregressive system with application to blind source separation," in *Proc. LVA/ICA*, 2010, pp. 245–253.
[6] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 9, pp. 1622–1637, Sep. 2016.
[7] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation with independent low-rank matrix analysis," in *Audio Source Separation*, S. Makino, Ed. Cham, Switzerland: Springer, 2018, pp. 125–155.
[8] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 31–35.
[9] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Interspeech*, Sep. 2016, pp. 545–549.

[10] D. Yu, M. Kolbaek, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 241–245.
[11] M. Kolbaek, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.
[12] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.
[13] Y. Liu and D. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2092–2102, Dec. 2019.
[14] J. Le Roux, G. Wichern, S. Watanabe, S. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 370–382, May 2019.
[15] M. Delfarah and D. Wang, "Deep learning for talker-dependent reverberant speaker separation: An empirical study," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 11, pp. 1839–1848, Nov. 2019.
[16] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1–5.
[17] T. Higuchi, K. Kinoshita, M. Delcroix, K. Žmolíková, and T. Nakatani, "Deep clustering-based beamforming for separation with unknown number of sources," in *Proc. Interspeech*, Aug. 2017, pp. 1183–1187.
[18] L. Drude and R. Haeb-Umbach, "Tight integration of spatial and spectral features for BSS with deep clustering embeddings," in *Proc. Interspeech*, Aug. 2017, pp. 2650–2654.
[19] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 9, pp. 1652–1664, Sep. 2016.
[20] 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. Audio, Speech, Lang. Process.*, vol. 27, no. 10, pp. 1601–1615, Oct. 2019.
[21] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Comput.*, vol. 31, no. 9, pp. 1891–1914, Sep. 2019.
[22] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 716–720.
[23] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE 28th Int. Workshop Mach. Learn. for Signal Process. (MLSP)*, Sep. 2018, pp. 1–6.
[24] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NIPS*, 2014, pp. 3581–3589.
[25] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2011, pp. 189–192.
[26] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 9, pp. 1432–1443, Sep. 2019.
[27] L. Li, H. Kameoka, and S. Makino, "Fast MVAE: Joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 546–550.
[28] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.
[29] C. Févotte and J. F. Cardoso, "Maximum likelihood approach for blind audio source separation using time-frequency Gaussian source models," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2005, pp. 78–81.
[30] E. Vincent, S. Arberet, and R. Gribonval, "Underdetermined instantaneous audio source separation via local Gaussian modeling," in *Proc. Integr. Comput.-Aided Eng.*, 2009, pp. 775–782.
[31] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2001, pp. 556–562.
[32] A. Ozerov and H. Kameoka, "Gaussian model based multichannel separation," in *Audio Source Separation and Speech Enhancement*, E. Vincent, T. Virtanen, S. Gannot, Eds. Springer, pp. 289–315, 2018.
[33] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
[34] S. Seki, H. Kameoka, L. Li, T. Toda, and K. Takeda, "Underdetermined source separation based on generalized multichannel variational autoencoder," *IEEE Access*, vol. 7, pp. 168104–168115, 2019.
[35] K. Sekiguchi, Y. Bando, A. A. Nugraha, K. Yoshii, and T. Kawahara, "Semi-supervised multichannel speech enhancement with a deep speech prior," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2197–2212, Dec. 2019.
[36] S. Leglaive, U. Simsekli, A. Liutkus, L. Girin, and R. Horaud, "Speech enhancement with variational autoencoders and alpha-stable distributions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 541–545.
[37] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 101–105.
[38] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders," in *Proc. Interspeech*, Sep. 2019, pp. 3158–3162.
[39] S. Inoue, H. Kameoka, L. Li, S. Seki, and S. Makino, "Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 96–100.
[40] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," in *Proc. Interspeech*, Aug. 2017, pp. 3642–3646.
[41] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Semi-blind source separation with multichannel variational autoencoder," 2018, *arXiv:1808.00892*. [Online]. Available: http://arxiv.org/abs/1808.00892
[42] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian multichannel speech enhancement with a deep speech prior," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 1233–1239.
[43] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Boca Raton, FL, USA: CRC Press, 1995.
[44] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. NIPS*, 2016, pp. 2172–2180.
[45] D. Barber and F. V. Agakov, "The IM algorithm: A variational approach to information maximization," in *Proc. NIPS*, 2003, pp. 1–8.
[46] S. Mogami, H. Sumino, D. Kitamura, N. Takamune, S. Takamichi, H. Saruwatari, and N. Ono, "Independent deeply learned matrix analysis for multichannel audio source separation," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1571–1575.
[47] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017, pp. 933–941.
[48] L. Li and H. Kameoka, "Deep clustering with gated convolutional networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 16–20.
[49] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-Sequence voice conversion with similarity metric learned using generative adversarial networks," in *Proc. Interspeech*, Aug. 2017, pp. 1283–1287.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[51] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701
[52] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," 2018, *arXiv:1804.04262*. [Online]. Available: http://arxiv.org/abs/1804.04262
[53] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, 1979.
[54] M. R. Schroeder, "New method of measuring reverberation time," *J. Acoust. Soc. Amer.*, vol. 37, no. 3, pp. 409–412, Mar. 1965.
[55] S. Nakamura, K. Hiyane, F. Asano, and T. Endo, "Sound scene data collection in real acoustical environments," *J. Acoust. Soc. Jpn. (E)*, vol. 20, no. 3, pp. 225–231, 1999.

[56] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.
[57] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)–A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing. Proc. (Cat. No.01CH1)*, vol. 2, 2001, pp. 749–752.
[58] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 4214–4217.
[59] J. S. Garofolo, *CSR-I (WSJ0) Complete LDC93S6A*. Philadelphia, PA, USA: Linguistic Data Consortium, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S6A

LI LI (Student Member, IEEE) received the B.E. degree from the Shanghai University of Finance and Economics, China, in 2014, and the M.S. degree from the University of Tsukuba, Japan, in 2018, where she is currently pursuing the Ph.D. degree with the Graduate School. Since 2018, she has been a Research Fellow of the Japan Society of Promotion of Science. Her research interests include audio and speech signal processing, source separation, and machine learning. She received the 13th Student Presentation Award from the Acoustical Society of Japan and the second IEEE Signal Processing Society Tokyo Joint Chapter Student Award.

HIROKAZU KAMEOKA (Senior Member, IEEE) received the B.E., M.S., and Ph.D. degrees from The University of Tokyo, Japan, in 2002, 2004, and 2007, respectively. From 2011 to 2016, he was an Adjunct Associate Professor with The University of Tokyo. He is currently a Senior Distinguished Researcher with NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. He is also an Adjunct Associate Professor with the National Institute of Informatics. He is the author or coauthor of about 150 articles in journal articles and peer-reviewed conference proceedings. His research interests include audio, speech, and music signal processing, and machine learning. He has been a member of the IEEE Audio and Acoustic Signal Processing Technical Committee, since 2017, and a member of the IEEE Machine Learning for Signal Processing Technical Committee, since 2019. He has received 17 awards, including the IEEE Signal Processing Society 2008 SPS Young Author Best Paper Award. Since 2015, he has been an Associate Editor of the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

SHOTA INOUE received the B.S. degree in information technology from the University of Tsukuba, Japan, where he is currently pursuing the master's degree. His research interests include audio and speech signal processing, source separation, and machine learning.

SHOJI MAKINO (Fellow, IEEE) received the B.E., M.E., and Ph.D. degrees from Tohoku University, Japan, in 1979, 1981, and 1993, respectively.

He joined NTT, in 1981. He is currently a Professor with the University of Tsukuba. He is the author or coauthor of more than 200 papers in journals and conference proceedings. He is responsible for more than 150 patents. His research interests include adaptive filtering technologies, realization of acoustic echo cancellation, blind source separation of convolutive mixtures of speech, and acoustic signal processing for speech and audio applications.

Dr. Makino is an IEICE Fellow, a Board Member of the ASJ, and a Member of EURASIP. He received the ICA Unsupervised Learning Pioneer Award, in 2006, the IEEE MLSP Competition Award, in 2007, the IEEE SPS Best Paper Award, in 2014, the Achievement Award for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, in 2015, the Hoko Award of the Hattori Hokokai Foundation, in 2018, the Outstanding Contribution Award of the IEICE, in 2018, the Technical Achievement Award of the IEICE, in 2017 and 1997, the Outstanding Technological Development Award of the ASJ, in 1995, and eight Best Paper Awards. He was the Chair of SPS Audio and Acoustic Signal Processing Technical Committee, from 2013 to 2014 and the Blind Signal Processing Technical Committee of the IEEE Circuits and Systems Society, from 2009 to 2010. He was the General Chair of IWAENC 2018, WASPAA2007, IWAENC2003, and the Organizing Chair of ICA2003. He is the designated Plenary Chair of ICASSP2012. He has served on IEEE SPS Board of Governors, from 2018 to 2020, Technical Directions Board, from 2013 to 2014, Awards Board, from 2006 to 2008, Conference Board, from 2002 to 2004, and Fellow Evaluation Committee, from 2018 to 2020. He was a member of the IEEE Jack S. Kilby Signal Processing Medal Committee, from 2015 to 2018, and the James L. Flanagan Speech & Audio Processing Award Committee, from 2008 to 2011. He was an Associate Editor of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, from 2002 to 2005, and the *EURASIP Journal on Advances in Signal Processing*, from 2005 to 2012. He was a Guest Editor of the Special Issue of the *IEEE Signal Processing Magazine*, from 2013 to 2014. He was a Keynote Speaker at ICA2007, a Tutorial speaker at EMBC2013, Interspeech2011, and ICASSP2007. From 2009 to 2010, he was an IEEE SPS Distinguished Lecturer.