# A Flow-Based Deep Latent Variable Model for Speech Spectrogram Modeling and Enhancement

Aditya Arie Nugraha [1], Kouhei Sekiguchi [1], and Kazuyoshi Yoshii [1]

[1]Affiliation not available

October 30, 2023

**Abstract**

This paper describes a deep latent variable model of speech power spectrograms and its application to semi-supervised speech enhancement with a deep speech prior. By integrating two major deep generative models, a variational autoencoder (VAE) and a normalizing flow (NF), in a mutually-beneficial manner, we formulate a flexible latent variable model called the NF-VAE that can extract low-dimensional latent representations from high-dimensional observations, akin to the VAE, and does not need to explicitly represent the distribution of the observations, akin to the NF. In this paper, we consider a variant of NF called the generative flow (GF a.k.a. Glow) and formulate a latent variable model called the GF-VAE. We experimentally show that the proposed GF-VAE is better than the standard VAE at capturing fine-structured harmonics of speech spectrograms, especially in the high-frequency range. A similar finding is also obtained when the GF-VAE and the VAE are used to generate speech spectrograms from latent variables randomly sampled from the standard Gaussian distribution. Lastly, when these models are used as speech priors for statistical multichannel speech enhancement, the GF-VAE outperforms the VAE and the GF.

# A Flow-Based Deep Latent Variable Model for Speech Spectrogram Modeling and Enhancement

Aditya Arie Nugraha, *Member, IEEE,* Kouhei Sekiguchi, *Member, IEEE,* and Kazuyoshi Yoshii, *Member, IEEE*

***Abstract*—This paper describes a deep latent variable model of speech power spectrograms and its application to semi-supervised speech enhancement with a deep speech prior. By integrating two major deep generative models, a variational autoencoder (VAE) and a normalizing flow (NF), in a mutually-beneficial manner, we formulate a flexible latent variable model called the NF-VAE that can extract low-dimensional latent representations from high-dimensional observations, akin to the VAE, and does not need to explicitly represent the distribution of the observations, akin to the NF. In this paper, we consider a variant of NF called the generative flow (GF a.k.a. Glow) and formulate a latent variable model called the GF-VAE. We experimentally show that the proposed GF-VAE is better than the standard VAE at capturing fine-structured harmonics of speech spectrograms, especially in the high-frequency range. A similar finding is also obtained when the GF-VAE and the VAE are used to generate speech spectrograms from latent variables randomly sampled from the standard Gaussian distribution. Lastly, when these models are used as speech priors for statistical multichannel speech enhancement, the GF-VAE outperforms the VAE and the GF.**

***Index Terms*—deep generative model, variational autoencoder, normalizing flow, power spectrogram, speech enhancement**

## I. Introduction

PROBABILISTIC spectrogram models play an essential role in modern audio signal processing [1], [2]. Recent semi-supervised speech enhancement methods for noisy speech, for example, use a deep generative model that produces a high-dimensional speech power spectrogram from low-dimensional latent variables as a prior distribution of clean speech [3]–[7]. Such a generative model can be trained in an unsupervised manner from clean speech data in the variational autoencoder (VAE) framework [8]. In speech enhancement, the compact latent variables can be estimated efficiently by a Markov chain Monte Carlo (MCMC) method [9], such as the Metropolis algorithm [10] and the Metropolis-Hastings algorithm [11], as shown in [3]–[7]. As recently proposed, besides the MCMC methods, the latent variables can also be estimated by back-propagation [12] to maximize the likelihood function as in [13] or by utilizing the recognition model [14].

Manuscript received XXX YYY, 2019; revised XXX YYY, 2019; accepted XXX YYY, 2020. Date of publication XXX YYY, 2020; date of current version XXX YYY, 2020. This work was partially supported by JSPS KAKENHI No. 19H04137 and NII CRIS-Line Collaborative Research. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhijian Ou. *(Corresponding author: Aditya Arie Nugraha.)*

The authors are with the Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo 103-0027, Japan (e-mail: {adityaarie.nugraha, kouhei.sekiguchi, kazuyoshi.yoshii}@riken.jp).

K. Sekiguchi and K. Yoshii are also with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.

Digital Object Identifier 10.1109/TASLP.2020.2979603

A major limitation of the VAE framework lies in imprecise (lossy) recognition and generation processes. Its generation process outputs tend to lose some details [15]–[17]. This would be problematic for speech spectrograms, in particular, because the harmonic structures should be preserved well. Additionally, the distribution of observations (spectrograms) has to be explicitly defined. It is not always easy to do so because in many real cases, we do not know whether a certain distribution is suitable for some observations and assuming an inappropriate one might result in a detrimental effect.

By contrast, the normalizing flow (NF) [18], [19] provides precise (lossless) recognition and generation processes due to its sequence of bijective transformations. Additionally, the distribution of the observations does not need to be explicitly defined, so NF could be used to model sophisticated ones. However, the dimensionality of the transformed variables is the same as that of the observations due to the bijective property. Therefore, the transformed variables might be high dimensional and their updates, especially by an MCMC method, would be computationally expensive. Moreover, updating using the recognition model [14] cannot be done because of the bijective property.

Considering the complementary properties of both VAE and NF, in this paper, we propose their combination as NF-VAE in which a VAE is used to discover low-dimensional latent variables from high-dimensional transformed variables obtained by NF[1]. In other words, instead of directly modeling the observations that possibly follow a sophisticated distribution, the VAE part of the NF-VAE models the transformed observations (the high-dimensional transformed variables) obtained by the GF part. It is worth noting that only a part of the NF-VAE preserves the bijective property and thus, the NF-VAE as a whole does not have precise recognition and generation processes. However, since the distribution of the transformed variables is chosen to be a simple one, we expect that the VAE part can accurately model those transformed variable so that the NF-VAE has better recognition and generation processes than the standalone VAE. The VAE part of the NF-VAE can be seen as a dimensionality reduction to mitigate the overfitting problem experienced by the NF and its variants due to the high-dimensional transformed variables. The low-dimensional latent variables in the NF-VAE limit the model capacity, which improves the generalization of the generation process. The use of low-dimensional latent variables also allows us to do an ef-

[1]Literally, the NF's transformed variables and the VAE's latent variables are *latent* because both are not observed. The NF's transformed variables can be exactly computed given the observations so that they have a deterministic nature and thus, they are not *latent variables* in a statistical sense.

ficient sampling. In summary, the proposed NF-VAE needs no explicit definition of the distribution of the observations similarly to the NF and allows low-dimensional latent variables as in the VAE. Additionally, in this paper, we consider a variant of NF called the generative flow (GF a.k.a. Glow) [20] and propose its combination with a VAE named the GF-VAE.

We introduce the formulations of VAE, NF, and NF-VAE in Section II. We then describe the applications of VAE, GF, and GF-VAE as deep generative models of speech power spectrograms (hereafter referred to as *deep speech models*) in Section III. We review a state-of-the-art semi-supervised multichannel speech enhancement method [7], in which the deep speech models are used to provide the speech prior probability, in Section IV. Afterwards, in Section V, we present the evaluation of the models on a clean speech reconstruction task, a random speech generation task, and a speech enhancement task. We show that the GF-VAE can produce better details of speech harmonic structures, especially in the high-frequency range, than the VAE. We also show that the GF-VAE outperforms the GF and the VAE for the aforementioned speech enhancement method. Finally, Section VI concludes this paper.

## II. Deep Generative Models

In this section, we briefly review the variational autoencoder (VAE) [8] and the normalizing flow (NF) [18], [19], and then propose the NF-VAE. Note that the VAE and the NF-VAE are latent variable models, but the NF is not. We also briefly discuss several key related works.

### A. Variational Autoencoder

Let us assume that an observed variable vector $\mathbf{x} \in \mathbb{R}^F$ is generated by a latent variable vector $\mathbf{z} \in \mathbb{R}^D$ with $D < F$. Following the variational inference principle [21], the log-likelihood of $\mathbf{x}$ is expressed as

$$\begin{aligned}\ln\, p(\mathbf{x}) &= \ln \int_{\mathbf{z}} p(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \ln \int_{\mathbf{z}} \frac{q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} p(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}\left[\ln\, \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] \\ &= \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}\left[\ln\, p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})), \end{aligned} \quad (1)$$

where $\mathbb{E}[\cdot]$ is the expectation and $\mathrm{KL}[q\|p]$ is the Kullback-Leibler divergence from $p$ to $q$ [22]. The latent variable vector is typically assumed to follow a simple distribution, e.g., the standard Gaussian distribution $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ [8], where $\mathbf{I}$ is the identity matrix, or the hyperspherical uniform distribution [23]. The variational posterior distribution $q(\mathbf{z}|\mathbf{x})$ is then defined accordingly. Most importantly, the observed variable distribution $p(\mathbf{x}|\mathbf{z})$ has to be defined appropriately.

Let $\theta$ and $\phi$ be two separate sets of deep neural network (DNN) parameters. A VAE can be seen to be composed of an encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ depicted as

$$\mathbf{x} \underset{\text{decoder}}{\overset{\text{encoder}}{\rightleftharpoons}} \mathbf{z}.$$

The encoder acts as a recognition model and the decoder acts as a generative model. It is worth mentioning that in the VAE framework, the recognition model is initially introduced to allow the generative model to be trained.

### B. Normalizing Flow

Let $\mathbf{x} \in \mathbb{R}^F$ and $\mathbf{y} \in \mathbb{R}^F$ be an observed variable vector and a transformed variable vector, respectively. The flow between $\mathbf{x}$ and $\mathbf{y}$ can be depicted as

$$\mathbf{x} \overset{g_1}{\longleftrightarrow} \mathbf{h}_1 \overset{g_2}{\longleftrightarrow} \cdots \overset{g_{K-1}}{\longleftrightarrow} \mathbf{h}_{K-1} \overset{g_K}{\longleftrightarrow} \mathbf{y},$$

where $K$ is the number of flow steps and indexed by $k$, $\mathbf{y} = g(\mathbf{x}) = g_K \circ g_{K-1} \circ \cdots \circ g_1(\mathbf{x})$, $\mathbf{h}_k = g_k(\mathbf{h}_{k-1})$, $\mathbf{h}_0 \triangleq \mathbf{x}$, $\mathbf{h}_K \triangleq \mathbf{y}$, and $\mathbf{x} = g^{-1}(\mathbf{y})$.

According to the change-of-variables principle [24], the log-likelihood of the observed variables $\mathbf{x}$ can be expressed as

$$\begin{aligned}\ln\, p(\mathbf{x}) &= \ln\, p(\mathbf{y}) \left|\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}}\right| \\ &= \ln\, p(\mathbf{y}) + \sum_{k=1}^{K} \ln \left|\frac{\mathrm{d}\mathbf{h}_k}{\mathrm{d}\mathbf{h}_{k-1}}\right| \\ &= \ln\, p(\mathbf{y}) + \sum_{k=1}^{K} \ln \left|\frac{\mathrm{d}g_k(\mathbf{h}_{k-1})}{\mathrm{d}\mathbf{h}_{k-1}}\right|, \end{aligned} \quad (2)$$

where $\left|\frac{\mathrm{d}g_k(\mathbf{h}_{k-1})}{\mathrm{d}\mathbf{h}_{k-1}}\right|$ is the Jacobian matrix determinant of $g_k$ evaluated at $\mathbf{h}_{k-1}$. The distribution of transformed variables is typically assumed to be a simple one, e.g., the standard Gaussian distribution $p(\mathbf{y}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

There exists different designs of flow step $g_k$ [18]–[20], [25], [26]. In this paper, we mainly follow the flow step design of the generative flow (GF a.k.a. Glow) [20], where a flow step consists of an activation normalization, a feature map permutation by an invertible $1\times1$ convolution, and an affine coupling layer, in which a DNN is used. We briefly describe each of these components in Section III-B and refer to [20] for further details. Let $\psi$ be a set of parameters gathering those of the aforementioned activation normalization, invertible $1\times1$ convolution, and DNN of all flow steps. The flow $\mathbf{y} = g_\psi(\mathbf{x})$ corresponds to the recognition process and the reverse flow $\mathbf{x} = g_\psi^{-1}(\mathbf{y})$ corresponds to the generation process.

### C. NF-VAE

Let us now considers an observed variable vector $\mathbf{x} \in \mathbb{R}^F$, a transformed variable vector $\mathbf{y} \in \mathbb{R}^F$, and a latent variable vector $\mathbf{z} \in \mathbb{R}^D$ with $D < F$. Building upon the NF and the VAE, we propose a new generative model named the NF-VAE and depicted as

$$\mathbf{x} \overset{g_1}{\longleftrightarrow} \mathbf{h}_1 \overset{g_2}{\longleftrightarrow} \cdots \overset{g_{K-1}}{\longleftrightarrow} \mathbf{h}_{K-1} \overset{g_K}{\longleftrightarrow} \mathbf{y} \underset{\text{decoder}}{\overset{\text{encoder}}{\rightleftharpoons}} \mathbf{z}.$$

As shown above, the NF-VAE consists of an NF part and a VAE part, composed of an encoder and a decoder. The NF part aims to transform the observed variable vector $\mathbf{x}$ following some unknown, possibly sophisticated, distribution into the transformed variable vector $\mathbf{y}$ with a known distribution, e.g., a Gaussian distribution. The VAE part then aims to find the low-dimensional latent variable vector $\mathbf{z}$ for the high-dimensional

transformed variable vector $\mathbf{y}$, which ultimately represents the high-dimensional observed variable vector $\mathbf{x}$. Beside preferable from a computational cost point of view, a low-dimensional representation with an appropriate size would capture essential underlying characteristics, e.g., pitch and timbre in the context of speech. The use of a low-dimensional representation can also be seen as a way to limit the model capacity so that the model has a good generalization capability. Because of this dimensionality reduction, the NF-VAE does not have precise recognition and generation processes, unlike the NF. However, we expect that the VAE part can model $\mathbf{y}$ accurately because the distribution of $\mathbf{y}$ is chosen so that it is a relatively simple one. Given an accurate $\mathbf{y}$, an accurate $\mathbf{x}$ could then be reconstructed.

Given the NF-VAE, a lower bound on the log-likelihood of $\mathbf{x}$ is obtained by combining (1) and (2):

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{y})}\left[\ln p(\mathbf{y}|\mathbf{z})\right] - \mathrm{KL}(q(\mathbf{z}|\mathbf{y})\|p(\mathbf{z})) + \sum_{k=1}^{K} \ln \left| \frac{\mathrm{d}g_k(\mathbf{h}_{k-1})}{\mathrm{d}\mathbf{h}_{k-1}} \right|. \quad (3)$$

We use the standard Gaussian distribution for the latent variable prior $p(\mathbf{z})\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ and a Gaussian distribution for the variational posterior $q(\mathbf{z}|\mathbf{y})$, as in the VAE. We also assume that the transformed variable distribution $p(\mathbf{y}|\mathbf{z})$ is a Gaussian distribution. Unlike the VAE, for the NF-VAE, we do not need to define the observed variable distribution.

Let $\theta$, $\phi$, and $\psi$ be the DNN parameters of the decoder, the encoder, and the NF part, respectively, as previously considered in Sections II-A and II-B. In general, the recognition process of NF-VAE involves a flow process and then an encoding as follows: $\widehat{\mathbf{z}} \sim q_\phi(\mathbf{z}|g_\psi(\mathbf{x}))$. Conversely, the generation process of NF-VAE involves a decoding $\widehat{\mathbf{y}} \sim p_\theta(\mathbf{y}|\mathbf{z})$ and then a reverse flow process $\widehat{\mathbf{x}} = g_\psi^{-1}(\widehat{\mathbf{y}})$. Since we adopt the flow step of the GF, we introduce a variant of NF-VAE named the GF-VAE in Section III-C.

### D. Related Work

Another notable model that combines a flow-based model and a VAE is presented in [18]. It employs an NF between the encoder and the decoder of the VAE. The latent variable estimated by the encoder might follow a sophisticated distribution, but the decoder input would follow a simple known distribution. In a similar spirit but with a much more advanced network design, the ResNet VAE in [26] uses an inverse autoregressive flow for estimating the latent variable posterior distribution at different layers. These models thus address the difficulty of choosing the posterior distribution of the latent variables and ultimately, improve its estimation. Our NF-VAE further addresses the difficulty of choosing the distribution of the observations.

Concurrently to our work, the dimensionality reduction flows (DRF) [27] has been introduced in the context of image generation. This DRF is fundamentally the same as our proposed NF-VAE. To the best of our knowledge, our proposal is still the first that considers the application of the model to speech generation and further, to a downstream task, i.e., a semi-supervised multichannel speech enhancement.

In speech processing, the GF has been combined with the WaveNet [28] to build the WaveGlow [29] and the FloWaveNet [30] for speech synthesis. Time-domain speech signals are generated from random samples (transformed variables) and conditioned on the mel-spectrogram estimated from a text. Our GF-VAE works in the time-frequency domain so that we could plug it into various existing speech enhancement methods [1]. Most importantly, our GF-VAE has an ability of randomly generating satisfying observations without any conditioning.

## III. Deep Speech Models

We now specify the DNN architectures and their training cost functions. Let us introduce the time frame index $t$ for the observed vectors, the transformed vectors, and the latent variable vectors with $T$ denotes the number of time frames in a training minibatch. Let speech power spectrum $\mathbf{x}_t \in \mathbb{R}_+^F$ be the observed variable vector whose dimension $F$ corresponds to the number of frequency bins. In this paper, we set $F = 513$ (see Section V-A2 for the description of the spectrum extraction).

### A. Variational Autoencoder

We assume that the latent variables follow the simple, standard Gaussian distribution $p(\mathbf{z}_t)\sim\mathcal{N}(\mathbf{0},\mathbf{I})$. We also assume $q_\phi(\mathbf{z}_t|\mathbf{x}_t)\sim\mathcal{N}\big(\boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t), \mathrm{diag}\big(\boldsymbol{\sigma}_\phi^{\mathbf{z}}(\mathbf{x}_t)^2\big)\big)$, where $\mathrm{diag}\big(\cdot\big)$ returns a diagonal matrix from a vector.

For speech enhancement, the speech complex spectrum at each time-frequency (TF) bin $(t, f)$ is typically assumed to follow a univariate complex-valued circularly-symmetric Gaussian distribution [31, Thm. 3.7.14] $\widetilde{x}_{ft}\sim\mathcal{N}_{\mathbb{C}}(0,\nu_{ft})$. Maximizing $\ln p(\widetilde{x}_{ft})$ for this distribution is equivalent (up to a constant) to maximizing $\ln p(|\widetilde{x}_{ft}|^2)$ for $|\widetilde{x}_{ft}|^2\sim\mathrm{Exp}(\lambda_{ft})$ with $\lambda_{ft} = \nu_{ft}^{-1}$. Hence, to model the speech power spectrogram with a VAE, the observed variables are typically assumed to follow an exponential distribution $p_\theta(\mathbf{x}_t|\mathbf{z}_t)\sim\mathrm{Exp}\left(\boldsymbol{\lambda}_\theta(\mathbf{z}_t)\right)$ [3]–[7]. In this case, the training cost function to be minimized, which is the negative of (1), is expressed as

$$\begin{aligned}\mathcal{D}_{\theta,\phi}^{\text{VAE}} &\triangleq -\mathbb{E}_{\mathbf{z}_t\sim q_\phi(\mathbf{z}_t|\mathbf{x}_t)}\left[\ln p_\theta(\mathbf{x}_t|\mathbf{z}_t)\right] + \mathrm{KL}(q_\phi(\mathbf{z}_t|\mathbf{x}_t)\|p(\mathbf{z}_t)) \\ &\triangleq \mathcal{D}_{\theta,\phi}^{\text{pow}} + \mathcal{D}_{\phi}^{\text{reg}}, \end{aligned} \quad (4)$$

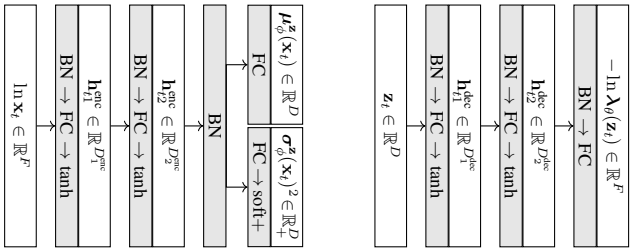
where

$$\begin{aligned}\mathcal{D}_{\theta,\phi}^{\text{pow}} &\triangleq \frac{1}{T}\sum_{f,t=1}^{F,T}\Big(\left\{\boldsymbol{\lambda}_\theta(\mathbf{z}_t)\right\}_f\left\{\mathbf{x}_t\right\}_f - \ln\left\{\boldsymbol{\lambda}_\theta(\mathbf{z}_t)\right\}_f\Big) \\ &= \frac{1}{T}\sum_{f,t=1}^{F,T}\Big(\left\{\boldsymbol{\lambda}_\theta(\mathbf{z}_{\phi,t})\right\}_f\left\{\mathbf{x}_t\right\}_f - \ln\left\{\boldsymbol{\lambda}_\theta(\mathbf{z}_{\phi,t})\right\}_f\Big), \end{aligned} \quad (5)$$

$$\begin{aligned}\mathcal{D}_{\phi}^{\text{reg}} \triangleq \frac{1}{2T}\sum_{d,t=1}^{D,T}\Big(&\left\{\boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t)\right\}_d^2 + \left\{\boldsymbol{\sigma}_\phi^{\mathbf{z}}(\mathbf{x}_t)\right\}_d^2 \\ &- \ln\left\{\boldsymbol{\sigma}_\phi^{\mathbf{z}}(\mathbf{x}_t)\right\}_d^2 - 1\Big), \end{aligned} \quad (6)$$

$$\mathbf{z}_{\phi,t} \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t) + \epsilon\boldsymbol{\sigma}_\phi^{\mathbf{z}}(\mathbf{x}_t), \quad (7)$$

with $\{\cdot\}_i$ is the $i$-th element of a vector and $\mathbf{z}_{\phi,t}$ is a random sample from $q_\phi(\mathbf{z}_t|\mathbf{x}_t)$ obtained with the reparameterization trick [8], whose $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, as shown in (7). The training optimizes parameters $\theta$ and $\phi$ simultaneously.




(a) Encoder $q_\phi(\mathbf{z}_t|\mathbf{x}_t)$ (b) Decoder $p_\theta(\mathbf{x}_t|\mathbf{z}_t)$

Fig. 1. Architectures of the encoder and the decoder for our VAE-2L model. 'BN' is the batch normalization [32], 'FC' is a fully-connected layer, 'tanh' is the hyperbolic tangent function, and 'soft+' is the softplus function.

We consider two VAEs, i.e., VAE-2L and VAE-3L, as the baseline models. As shown in Fig. 1, the VAE-2L has two intermediate layers in each encoder and decoder. The dimensions are $D_1^{\text{enc}} = D_2^{\text{dec}} = 512$, $D_2^{\text{enc}} = D_1^{\text{dec}} = 128$, and $D \in \{8, 16, 32\}$. By contrast, the VAE-3L has three intermediate layers in each encoder and decoder so that we have $h_{t3}^{\text{enc}}$ and $h_{t3}^{\text{dec}}$ after $h_{t2}^{\text{enc}}$ and $h_{t2}^{\text{dec}}$, respectively. The dimensions are $D_1^{\text{enc}} = D_3^{\text{dec}} = 384$, $D_2^{\text{enc}} = D_2^{\text{dec}} = 256$, and $D_3^{\text{enc}} = D_1^{\text{dec}} = 128$. The total number of parameters for both VAE-2L and VAE-3L is about 665k. For random generation, we sample $\mathbf{z}_t$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to obtain the scale of an exponential distribution $\boldsymbol{\lambda}_\theta(\mathbf{z}_t)$ using the decoder. Instead of sampling $\widehat{\mathbf{x}}_t$ from the distribution $\text{Exp}\left(\boldsymbol{\lambda}_\theta(\mathbf{z}_t)\right)$, for practical purposes, we opt to take the expected value of that distribution $\widehat{\mathbf{x}}_t \triangleq \boldsymbol{\lambda}_\theta(\mathbf{z}_t)^{-1}$.

### *B. Generative Flow*

We use a multi-scale architecture [19] shown in Fig. 2, in which parts of the transformed variables are extracted after different flow steps. We apply a replicative padding at the high-frequency side of $\mathbf{x}_t$ to obtain $\bar{\mathbf{x}}_t \in \mathbb{R}^{\bar{F}}$ with $\bar{F} = 528$. The *squeeze* function splits the spectrogram along the frequency axis into $C_0 = 16$ feature maps, whose height is $\tilde{F} = 33$, so that we have $\mathbf{H}_{t0} \in \mathbb{R}^{C_0 \times \tilde{F}}$. This $\mathbf{H}_{t0}$ is then processed through the flow with $K = 16$. Each flow step $k$ results in $\mathbf{H}_{tk} \in \mathbb{R}^{C_k \times \tilde{F}}$. After every two flow steps, we first partition $\mathbf{H}_{tk'}$ into $\mathbf{Y}'_{tk'} \in \mathbb{R}^{2 \times \tilde{F}}$ and $\mathbf{H}'_{tk'} \in \mathbb{R}^{(C_k - 2) \times \tilde{F}}$, where $k' \in [1, K] \wedge k' \in 2\mathbb{N}$, and then extract $\mathbf{Y}'_{tk'}$ as a part of the padded and reshaped transformed variables $\mathbf{Y}_t \in \mathbb{R}^{C_0 \times \tilde{F}}$. The following flow step after each extraction processes $\mathbf{H}'_{tk'}$. Therefore, different flow steps may process different numbers of feature maps. The number of feature maps processed at the flow step $k$ is $C_k = C_0 - 2\left((k-1)//2\right)$, where // denotes a floor division. We gather and concatenate all of the transformed variable parts at the end of the flow to form $\mathbf{Y}_t = \{\mathbf{Y}'_{tk'} \,|\, \forall k' \in [1, K] \wedge k' \in 2\mathbb{N}\}$ and apply the *unsqueeze* function on it to construct $\bar{\mathbf{y}}_t \in \mathbb{R}^{\bar{F}}$. While the padding is kept during the training phase, in the test phase, the padding is then removed from $\bar{\mathbf{y}}_t$ to finally obtain the transformed variables $\mathbf{y}_t$.

Following the design in [20], one flow step $g_k$ consists of an activation normalization layer, a feature map permutation layer by an invertible $1 \times 1$ convolution, and an affine coupling layer. This flow step design is depicted in Fig. 3(a). The parameters of the $k$-th activation normalization layer are the scale $\mathbf{s}_k^{\text{act}} \in \mathbb{R}^{C_k}$ and the bias $\mathbf{b}_k^{\text{act}} \in \mathbb{R}^{C_k}$. The parameters of the $k$-th feature permutation layer is the weight of the convolution kernel $\mathbf{W}_k \in \mathbb{R}^{C_k \times C_k}$. Each affine coupling layer has a single multilayer perceptron (MLP), parameterized by $\rho_k$, which is used for every feature maps $c \in [1, \frac{1}{2}C_k]$. The MLP architecture is illustrated in Fig. 3(b). We first split the affine coupling layer input $\bar{\mathbf{H}}_{tk} \in \mathbb{R}^{C_k \times \tilde{F}}$ into $\mathbf{H}_{tk}^{\text{a}}, \mathbf{H}_{tk}^{\text{b}} \in \mathbb{R}^{\frac{1}{2}C_k \times \tilde{F}}$. Let $\{\cdot\}_{i\bullet}$ be the $i$-th row of a matrix and $\{\cdot\}_{ij}$ be the $j$-th element of that $i$-th row. The MLP takes as input $\{\mathbf{H}_{tk}^{\text{a}}\}_{c\bullet}$ and returns, essentially, the affine transform parameters $\{\mathbf{S}_{\rho_k,tk}^{\text{aff}}\}_{c\bullet}$ and $\{\mathbf{B}_{\rho_k,tk}^{\text{aff}}\}_{c\bullet}$ to be applied to $\{\mathbf{H}_{tk}^{\text{b}}\}_{c\bullet}$ as

$$\{\mathbf{H}_{tk}^{\text{b}\prime}\}_{c\bullet} = \{\mathbf{S}_{\rho_k,tk}^{\text{aff}}\}_{c\bullet} \odot \{\mathbf{H}_{tk}^{\text{b}}\}_{c\bullet} + \{\mathbf{B}_{\rho_k,tk}^{\text{aff}}\}_{c\bullet}, \quad (8)$$

where $\odot$ denotes the Hadamard product. We then concatenate $\mathbf{H}_{tk}^{\text{a}}$ and $\mathbf{H}_{tk}^{\text{b}\prime}$ to obtain the output of the $k$-th affine coupling layer $\mathbf{H}_{tk}$, which is also the output of the $k$-th flow step.

As in the VAE, we assume that the transformed variables follow the simple, standard Gaussian distribution $p(\mathbf{y}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The training cost function, which is the negative of (2), for optimizing the parameters $\psi = \{\mathbf{s}_k^{\text{act}}, \mathbf{b}_k^{\text{act}}, \mathbf{W}_k, \rho_k \,|\, \forall k \in [1, K]\}$ is expressed as

$$\begin{aligned}\mathcal{D}_\psi^{\text{GLW}} &\triangleq -\ln p(\mathbf{y}_t) - \sum_{k=1}^{K} \ln \left| \frac{\mathrm{d} g_{\psi,k}(\mathbf{h}_{k-1})}{\mathrm{d}\mathbf{h}_{k-1}} \right| \\ &\triangleq \mathcal{D}_\psi^{\text{rp0}} + \sum_{k=1}^{K} \mathcal{D}_{\psi,k}^{\text{flw}}, \end{aligned} \quad (9)$$

where

$$\begin{aligned}\mathcal{D}_\psi^{\text{rp0}} &\triangleq \frac{1}{2T} \sum_{f,t=1}^{\bar{F},T} \left( \ln 2\pi + \{\mathbf{y}_t\}_f^2 \right) \\ &= \frac{1}{2T} \sum_{f,t=1}^{\bar{F},T} \left( \ln 2\pi + \{g_\psi(\mathbf{x}_t)\}_f^2 \right), \end{aligned} \quad (10)$$

$$\begin{aligned}\mathcal{D}_{\psi,k}^{\text{flw}} \triangleq &- \sum_{c=1}^{C_k} \ln \left| \{\mathbf{s}_k^{\text{act}}\}_c \right| - \frac{1}{T} \sum_{c,f,t=1}^{\frac{1}{2}C_k,\tilde{F},T} \ln \left| \{\mathbf{S}_{\rho_k,tk}^{\text{aff}}\}_{cf} \right| \\ &- \tilde{F} \ln |\mathbf{W}_k| . \end{aligned} \quad (11)$$

The parameters $\psi$ are initialized as in [20]. The parameters $\mathbf{s}_k^{\text{act}}, \mathbf{b}_k^{\text{act}}$ are initialized based on the first minibatch so that each feature map has zero mean and unit variance. The parameter $\mathbf{W}_k$ is initialized with a random orthonormal matrix. The parameter $\rho_k$ is initialized so that the affine coupling layer performs an identity function. We refer to [19], [20] for further details, including the Jacobian matrix determinants for the activation normalization, the invertible $1 \times 1$ convolution, and the affine coupling layer, that compose $\mathcal{D}_{\psi,k}^{\text{flw}}$ in (11). The total number of parameters in our GF model is about 30k. For random generation, we sample $\mathbf{y}_t$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and perform the reverse flow to obtain $\widehat{\mathbf{x}}_t = g_\psi^{-1}(\mathbf{y}_t)$.

### *C. GF-VAE*

For the GF-VAE, as for the VAE above, we assume that $p(\mathbf{z}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $q_\phi(\mathbf{z}_t|\mathbf{y}_t) \sim \mathcal{N}\big(\boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{y}_t), \text{diag}\big(\boldsymbol{\sigma}_\phi^{\mathbf{z}}(\mathbf{y}_t)^2\big)\big)$. We then consider two variants of GF-VAE with different assumptions of the transformed variable distribution:

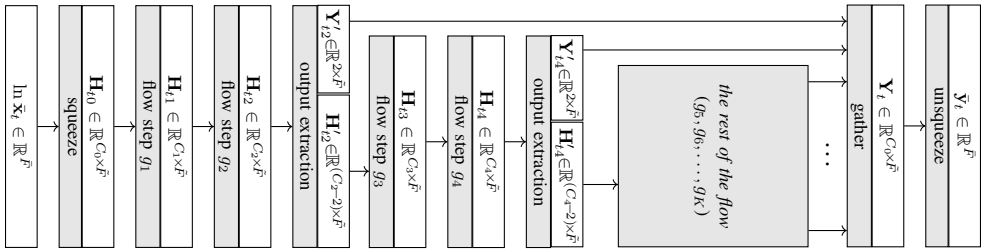
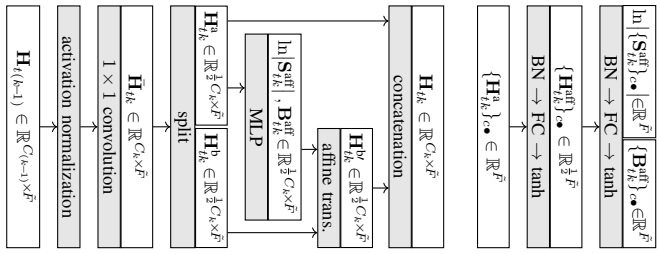
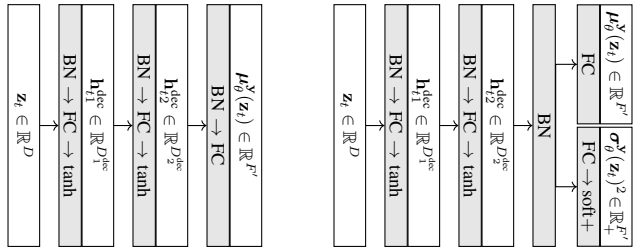



Fig. 2. Block diagram of our GF model. Only the first four flow steps $g_1, g_2, g_3, g_4$ are shown. The rest of the flow consisting of the flow steps $g_5, g_6, \ldots, g_K$ has the same construction.




(a) A single flow step $g_k$ (b) MLP

Fig. 3. Block diagram of a single flow step $g_k$ and the MLP whose outputs are used for the affine transformation in the flow step. For the sake of legibility, the parameters $\rho_k$ of the MLP is not shown in the figures.




(a) Decoder GF-VAE-1 $p_\theta(\mathbf{y}_t|\mathbf{z}_t)$ (b) Decoder GF-VAE-2 $p_\theta(\mathbf{y}_t|\mathbf{z}_t)$

Fig. 4. Architectures of the decoders for our GF-VAE-1 and GF-VAE-2 models.

- GF-VAE-1: $p_\theta(\mathbf{y}_t|\mathbf{z}_t) \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t), \mathbf{I})$,
- GF-VAE-2: $p_\theta(\mathbf{y}_t|\mathbf{z}_t) \sim \mathcal{N}\big(\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t), \mathrm{diag}\big(\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_t)^2\big)\big)$.

It is worth emphasizing that the nature of transformed variable vectors $\mathbf{y}_t$ in the GF-VAE is different from that in the GF. In the GF, $\mathbf{y}_t$ is assumed to follow a standard Gaussian distribution $p(\mathbf{y}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By contrast, $\mathbf{y}_t$ in the GF-VAE is assumed to follow a Gaussian distribution parameterized by DNN outputs as shown above.

The training cost functions for GF-VAE-1 $\mathcal{D}^{\mathrm{VGL1}}_{\theta,\phi,\psi}$ and GF-VAE-2 $\mathcal{D}^{\mathrm{VGL2}}_{\theta,\phi,\psi}$, which are the negative of (3), are expressed as

$$\begin{aligned}\mathcal{D}^{\mathrm{VGL1/2}}_{\theta,\phi,\psi} &\triangleq -\mathbb{E}_{\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{y}_t)}\left[\ln\, p_\theta(\mathbf{y}_t|\mathbf{z}_t)\right] \\ &\quad + \mathrm{KL}(q_\phi(\mathbf{z}_t|\mathbf{y}_t)\|p(\mathbf{z}_t)) - \sum_{k=1}^{K} \ln\left|\frac{\mathrm{d}g_{\psi,k}(\mathbf{h}_{k-1})}{\mathrm{d}\mathbf{h}_{k-1}}\right| \\ &\triangleq \mathcal{D}^{\mathrm{rp1/2}}_{\theta,\phi,\psi} + \mathcal{D}^{\mathrm{reg}}_{\phi,\psi} + \sum_{k=1}^{K} \mathcal{D}^{\mathrm{flw}}_{\psi,k}, \end{aligned} \tag{12}$$

where

$$\begin{aligned}\mathcal{D}^{\mathrm{rp1}}_{\theta,\phi,\psi} &\triangleq \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\ln 2\pi + \Big(\{\mathbf{y}_t\}_f - \{\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t)\}_f\Big)^2\right) \\ &= \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\ln 2\pi + \Big(\{g_\psi(\mathbf{x}_t)\}_f - \{\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_{\phi,\psi,t})\}_f\Big)^2\right), \end{aligned} \tag{13}$$

$$\begin{aligned}\mathcal{D}^{\mathrm{rp2}}_{\theta,\phi,\psi} &\triangleq \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\ln 2\pi + \ln\{\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_t)\}^2_f\right) + \\ &\quad \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\frac{\Big(\{\mathbf{y}_t\}_f - \{\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t)\}_f\Big)^2}{\{\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_t)\}^2_f}\right). \\ &= \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\ln 2\pi + \ln\{\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_{\phi,\psi,t})\}^2_f\right) + \\ &\quad \frac{1}{2T}\sum_{f,t=1}^{\bar{F},T}\left(\frac{\Big(\{g_\psi(\mathbf{x}_t)\}_f - \{\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_{\phi,\psi,t})\}_f\Big)^2}{\{\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_{\phi,\psi,t})\}^2_f}\right), \end{aligned} \tag{14}$$

$$\mathbf{z}_{\phi,\psi,t} \triangleq \boldsymbol{\mu}^{\mathbf{z}}_\phi(g_\psi(\mathbf{x}_t)) + \epsilon\boldsymbol{\sigma}^{\mathbf{z}}_\phi(g_\psi(\mathbf{x}_t)), \tag{15}$$

with $\mathcal{D}^{\mathrm{reg}}$ and $\mathcal{D}^{\mathrm{flw}}_{\psi,k}$ are shown in (6) and (11), respectively, and $\mathbf{z}_{\phi,\psi,t}$ is a random sample from $q_\phi(\mathbf{z}_t|\mathbf{y}_t)$ obtained with the reparameterization trick [8], whose $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, as shown in (15). The terms $\mathcal{D}^{\mathrm{rp1}}_{\theta,\phi,\psi}$ and $\mathcal{D}^{\mathrm{rp2}}_{\theta,\phi,\psi}$ are used for the GF-VAE-1 and the GF-VAE-2, respectively. The term $\mathcal{D}^{\mathrm{rp1}}_{\theta,\phi,\psi}$ is equivalent (up to a constant) to the widely used mean squared error. The training optimizes parameters $\theta$, $\phi$, and $\psi$ simultaneously.

The models are obtained by attaching a VAE at the transformed variable side of the GF shown in Fig. 2. The encoders of the GF-VAE-1 and GF-VAE-2 are similar to the one of VAE-2L shown in Fig. 1(a), but with $\mathbf{y}_t = g_\psi(\mathbf{x}_t) \in \mathbb{R}^{\bar{F}}$ as the input. The decoder of the GF-VAE-1 is similar to the one shown in Fig. 1(b), but it returns $\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t) \in \mathbb{R}^{\bar{F}}$ as shown in Fig. 4(a). Conversely, the output layers of the decoder of the GF-VAE-2 are similar to those of its encoder, i.e., the decoder outputs $\boldsymbol{\mu}^{\mathbf{y}}_\theta(\mathbf{z}_t) \in \mathbb{R}^{\bar{F}}$ and $\boldsymbol{\sigma}^{\mathbf{y}}_\theta(\mathbf{z}_t)^2 \in \mathbb{R}^{\bar{F}}_+$, as shown in Fig. 4(b). The GF-VAE-1 has $D^{\mathrm{enc}}_1 = D^{\mathrm{dec}}_2 = 480$ and $D^{\mathrm{enc}}_2 = D^{\mathrm{dec}}_1 = 120$, while the GF-VAE-2 has $D^{\mathrm{enc}}_1 = D^{\mathrm{dec}}_2 = 360$ and $D^{\mathrm{enc}}_2 = D^{\mathrm{dec}}_1 = 90$. The total numbers of parameters are about 669k and 679k for the GF-VAE-1 and the GF-VAE-2, respectively. Thus, our GF-VAEs are comparable in size to our VAEs above. For random generation, we sample $\mathbf{z}_t$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to

obtain $\boldsymbol{\mu}_\theta^{\mathbf{y}}(\mathbf{z}_t)$ using the decoder. We then use $\widehat{\mathbf{y}}_t \triangleq \boldsymbol{\mu}_\theta^{\mathbf{y}}(\mathbf{z}_t)$ to obtain $\widehat{\mathbf{x}}_t = g_\psi^{-1}(\widehat{\mathbf{y}}_t)$ using the reverse flow.

## IV. Speech Enhancement with Deep Speech Prior

In addition to the speech generation capability of the different deep speech models described above, we are also interested in their application to speech enhancement. In this section, we briefly review a state-of-the-art statistical semi-supervised multichannel speech enhancement method [7], that uses a pretrained deep generative model of speech as a prior distribution, and emphasize on how we integrate and use our VAE, GF, or GF-VAE based deep speech model. In this case, the deep speech model is also referred to as the deep speech prior. Although the method can handle one speech source and multiple noise sources, we consider one speech source and one noise source in this paper because this setting is known to work well. We refer to [7] for further details.

### A. Source and Spatial Modeling

Let $\widetilde{\mathbf{x}}_{ft}^{\mathrm{S}} \in \mathbb{C}^M$ and $\widetilde{\mathbf{x}}_{ft}^{\mathrm{N}} \in \mathbb{C}^M$ be an $M$-channel speech image and an $M$-channel noise image, respectively, at TF bin $(t, f)$. The observed multichannel mixture is expressed as

$$\widetilde{\mathbf{x}}_{ft} = \sum_{s \in \{\mathrm{S,N}\}} \widetilde{\mathbf{x}}_{ft}^{s}. \tag{16}$$

Following the local Gaussian model [33], each TF bin is assumed to follow a multivariate complex-valued circularly-symmetric Gaussian distribution [31, Thm. 3.7.14]:

$$\widetilde{\mathbf{x}}_{ft}^{s} \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \mathbf{R}_{ft}^{s} = \nu_{ft}^{s}\mathbf{G}_{f}^{s}\right), \tag{17}$$

$$\widetilde{\mathbf{x}}_{ft} \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \mathbf{R}_{ft} = \sum_{s \in \{\mathrm{S,N}\}} \nu_{ft}^{s}\mathbf{G}_{f}^{s}\right), \tag{18}$$

where $\nu_{ft}^{s} \in \mathbb{R}_+$ is the source power spectral density (or spectrogram) and $\mathbf{G}_f^s \in \mathbb{S}_+^{M \times M}$ is the positive-definite source spatial covariance matrix.

The speech power spectrogram is parameterized as

$$\nu_{ft}^{\mathrm{S}} = u_f v_t \{\widehat{\mathbf{x}}_t\}_f, \tag{19}$$

where $u_f$ is a frequency-dependent scaling factor, $v_t$ is a time-dependent scaling factor, and $\{\widehat{\mathbf{x}}_t\}_f$ is a power spectrum estimate. Recall from Section III that $\{\widehat{\mathbf{x}}_t\}_f \triangleq \{\boldsymbol{\lambda}_\theta(\mathbf{z}_t)^{-1}\}_f$ for the VAE-based speech model, $\{\widehat{\mathbf{x}}_t\}_f \triangleq \{g_\psi^{-1}(\mathbf{y}_t)\}_f$ for the GF-based speech model, and $\{\widehat{\mathbf{x}}_t\}_f \triangleq \{g_\psi^{-1}(\boldsymbol{\mu}_\theta^{\mathbf{y}}(\mathbf{z}_t))\}_f$ for the GF-VAE-based speech model.

The noise power spectrogram is modeled by a nonnegative matrix factorization (NMF) [34] as

$$\nu_{ft}^{\mathrm{N}} = \sum_{l=1}^{L} w_{lf} h_{lt}, \tag{20}$$

where $L$ is the number of basis spectra indexed by $l$, $w_{lf} \in \mathbb{R}_+$ is the $l$-th basis spectrum, and $h_{lt} \in \mathbb{R}_+$ is the $l$-th activation.

Let $\boldsymbol{\Psi}$ be a parameter set composed of all parameters $u_f$, $v_t$, $w_{lf}$, $h_{lt}$, $\mathbf{G}_f^s$, and $\mathbf{y}_t$ or $\mathbf{z}_t$ (depends on the deep speech model). The log-likelihood function to be maximized is expressed as

$$\begin{aligned}\ln p(\mathbf{X}|\boldsymbol{\Psi}) &= \sum_{f,t=1}^{F,T} \ln \mathcal{N}_{\mathbb{C}}\left(\widetilde{\mathbf{x}}_{ft}|\mathbf{0}, \mathbf{R}_{ft}\right) \\ &= \sum_{f,t=1}^{F,T} \left(-\mathrm{tr}\left(\mathbf{R}_{ft}^{-1}\mathbf{R}_{ft}^{\mathbf{x}}\right) - \ln|\mathbf{R}_{ft}|\right) + \text{const.}, \end{aligned} \tag{21}$$

where $\mathbf{X}$ is a set composed of all $\widetilde{\mathbf{x}}_{ft}$ and $\mathbf{R}_{ft}^{\mathbf{x}} = \widetilde{\mathbf{x}}_{ft}\widetilde{\mathbf{x}}_{ft}^{\star}$ is the observed mixture covariance matrix. The parameters of the pretrained deep speech model are kept fixed. With this model, we want to obtain the optimal generative model input $\mathbf{y}_t$ or $\mathbf{z}_t$ that, together with the other parameters in $\boldsymbol{\Psi}$, maximizes the log-likelihood function (21).

### B. Parameter Initialization

Let $\mathbf{x}_t \triangleq \left[\frac{1}{M}\mathrm{tr}\left(\mathbf{R}_{1t}^{\mathbf{x}}\right), \frac{1}{M}\mathrm{tr}\left(\mathbf{R}_{2t}^{\mathbf{x}}\right), \ldots, \frac{1}{M}\mathrm{tr}\left(\mathbf{R}_{Ft}^{\mathbf{x}}\right)\right] \in \mathbb{R}_+^F$ be the observed power spectrogram vector, where $\mathrm{tr}(\cdot)$ is the trace of a matrix. When the VAE-based speech model is used, the speech latent variables are initialized with the encoder given the observed power spectrogram, $\mathbf{z}_t^{\text{init}} \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t)$. When the GF-based speech model is used, the speech transformed variables are initialized with the flow, $\mathbf{y}_t^{\text{init}} = g_\psi(\mathbf{x}_t)$. When the GF-VAE-based speech model is used, the speech latent variables are initialized with the flow and the encoder, $\mathbf{z}_t^{\text{init}} \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(g_\psi(\mathbf{x}_t))$. Note that although the deep speech models are trained on clean speech, these models take noisy speech as input for the parameter initialization, which results in non-optimal speech latent or transformed variables. However, this initialization with a recognition model arguably still results in better initial speech latent or transformed variables for a speech enhancement task than the random initialization by sampling from the corresponding prior distribution. The speech spectral scale parameters are initialized as $u_f = \frac{1}{F}$ and $v_t = 1$.

For the noise spectral parameters, the basis spectra $\mathbf{w}_l \triangleq \left[w_{l1}, w_{l2}, \ldots, w_{lF}\right]$ is initialized with random samples from a Dirichlet distribution [7, Eq. (78)] and the activation $h_{lt}$ is initialized with random samples from a Gamma distribution [7, Eq. (79)].

The speech spatial parameters are initialized based on the observed mixture and the noise spatial parameters are initialized as scaled identity matrices as follows:

$$\mathbf{G}_f^{\mathrm{S}} \leftarrow \frac{\sum_{t=1}^{T} \mathbf{R}_{ft}^{\mathbf{x}}}{\sum_{t=1}^{T} \mathrm{tr}(\mathbf{R}_{ft}^{\mathbf{x}})}, \tag{22}$$

$$\mathbf{G}_f^{\mathrm{N}} \leftarrow \frac{1}{M}\mathbf{I}. \tag{23}$$

### C. Parameter Update

The parameter update [7, Algorithm 1] is based on the minorization-maximization (MM) principle [35]. Specifically, we derive a lower bound function with auxiliary parameters that *minorizes* the log-likelihood function (21). We then iteratively *maximize* the lower bound function (i.e., alternately optimize the parameters of the likelihood function and the auxiliary

parameters introduced for the minorization) so that the log-likelihood function is maximized indirectly. The parameter update is based on multiplicative update rules given by (25)–(31). We refer to [7] for further details, including the auxiliary function formula and the derivation of the update rules.

In each parameter update iteration, we first update the speech spectral parameters $u_f$ and $v_t$, the noise spectral parameters $w_{lf}$ and $h_{lt}$, and the source spatial parameters $\mathbf{G}_f^s$. We then update the speech latent variables $\mathbf{z}_t$ (for the VAE-based or the GF-VAE-based speech model) or transformed variables $\mathbf{y}_t$ (for the GF-based speech model) using multiple iterations of the Metropolis sampling method [10]. At the end of each parameter update iteration, the parameters are normalized to satisfy the following constraints: $\sum_{f=1}^F u_f = 1$; $\sum_{f=1}^F w_{lf} = 1, \forall l$; and $\text{tr}(\mathbf{G}_f^s) = 1, \forall s, \forall f$. After the parameter update iterations, the final estimated multichannel speech image is obtained by multichannel Wiener filtering:

$$\widehat{\widetilde{\mathbf{x}}}_{ft}^{\text{S}} = \mathbf{R}_{ft}^{\text{S}} \mathbf{R}_{ft}^{-1} \widetilde{\mathbf{x}}_{ft}. \tag{24}$$

The update rules for the speech spectral parameters $u_f$ and $v_t$ are given by

$$u_f \leftarrow u_f \sqrt{\frac{\sum_{t=1}^T v_t \{\widehat{\mathbf{x}}_t\}_f \, \text{tr}\Big(\mathbf{G}_f^{\text{S}} \mathbf{R}_{ft}^{-1} \mathbf{R}_{ft}^{\mathbf{x}} \mathbf{R}_{ft}^{-1}\Big)}{\sum_{t=1}^T v_t \{\widehat{\mathbf{x}}_t\}_f \, \text{tr}\Big(\mathbf{G}_f^{\text{S}} \mathbf{R}_{ft}^{-1}\Big)}}, \tag{25}$$

$$v_t \leftarrow v_t \sqrt{\frac{\sum_{f=1}^F u_f \{\widehat{\mathbf{x}}_t\}_f \, \text{tr}\Big(\mathbf{G}_f^{\text{S}} \mathbf{R}_{ft}^{-1} \mathbf{R}_{ft}^{\mathbf{x}} \mathbf{R}_{ft}^{-1}\Big)}{\sum_{f=1}^F u_f \{\widehat{\mathbf{x}}_t\}_f \, \text{tr}\Big(\mathbf{G}_f^{\text{S}} \mathbf{R}_{ft}^{-1}\Big)}}. \tag{26}$$

The update rules for the noise spectral parameters $w_{lf}$ and $h_{lt}$ are given by

$$w_{lf} \leftarrow w_{lf} \sqrt{\frac{\sum_{t=1}^T h_{lt} \, \text{tr}\Big(\mathbf{G}_f^{\text{N}} \mathbf{R}_{ft}^{-1} \mathbf{R}_{ft}^{\mathbf{x}} \mathbf{R}_{ft}^{-1}\Big)}{\sum_{t=1}^T h_{lt} \, \text{tr}\Big(\mathbf{G}_f^{\text{N}} \mathbf{R}_{ft}^{-1}\Big)}}, \tag{27}$$

$$h_{lt} \leftarrow h_{lt} \sqrt{\frac{\sum_{f=1}^F w_{lf} \, \text{tr}\Big(\mathbf{G}_f^{\text{N}} \mathbf{R}_{ft}^{-1} \mathbf{R}_{ft}^{\mathbf{x}} \mathbf{R}_{ft}^{-1}\Big)}{\sum_{f=1}^F w_{lf} \, \text{tr}\Big(\mathbf{G}_f^{\text{N}} \mathbf{R}_{ft}^{-1}\Big)}}. \tag{28}$$

The update rule for the spatial parameters $\mathbf{G}_f^s$ is given by

$$\mathbf{A}_f^s = \sum_{t=1}^T \nu_{ft}^s \mathbf{R}_{ft}^{-1} \mathbf{R}_{ft}^{\mathbf{x}} \mathbf{R}_{ft}^{-1}, \tag{29}$$

$$\mathbf{B}_f^s = \sum_{t=1}^T \nu_{ft}^s \mathbf{R}_{ft}^{-1}, \tag{30}$$

$$\mathbf{G}_f^s \leftarrow \big(\mathbf{G}_f^s \mathbf{A}_f^s \mathbf{G}_f^s\big) \, \#(\mathbf{B}_f^s)^{-1}, \tag{31}$$

where $\mathbf{A}\#\mathbf{B} = \mathbf{A}(\mathbf{A}^{-1}\mathbf{B})^{\frac{1}{2}}$ is the geometric mean of two positive semidefinite matrices $\mathbf{A}$ and $\mathbf{B}$ [36], [37].

## V. Evaluation

This section discusses the behavior and performance comparison of the different models. Section V-B examines whether modeling the transformed variables with a VAE is easier than modeling the observed variables directly. The following sections evaluate the performance of the different models for speech generation and multichannel speech enhancement. We first evaluate whether a model can reconstruct a clean speech spectrogram in Section V-C. In this speech reconstruction task, a clean speech spectrogram is assumed to be available and used to estimate the oracle latent variables with a recognition model. Since the test data is unseen during the training, the latent variables here are *oracle* in a loose sense. The latent variables are then used to obtain a speech spectrogram with a generation model. This reconstruction scenario is useful to check whether our models, each consisting of a recognition model and a generation model, work well in a relatively ideal setting. However, it obviously has a very limited practical usage. In practice, we need to process noisy speech without any oracle information. For a speech enhancement task, we are interested in the generation capability of our models given non-oracle latent variables. Instead of *oracle* latent variables, Section V-D thus checks whether our generation models can generate a speech spectrogram from *random* latent variables sampled from their respective prior distributions. Afterwards, in Section V-E, we use our models for a speech enhancement task, in which those latent variables are iteratively optimized as described in Section IV-C. Audio samples are available online[2].

### A. Settings

*1) Dataset:* We use the simulated training, development, and test sets of the CHiME-3/4 corpus [38], [39] consisting of multichannel mixtures of speech and noise. All data are sampled at 16 kHz. We train all models on the training set (7138 utterances $\approx$ 15.0 hours) and validate them on the development set (1640 utterances $\approx$ 2.9 hours). For this purpose, we only use the clean speech from the microphone that directly faces the speaker (hereafter referred to as *the center front-facing microphone*). The evaluation is done on the test set (1320 utterances $\approx$ 2.3 hours). The speech reconstruction task employs single-channel clean speech of the center front-facing microphone as in the model training, while the speech enhancement task considers 5-channel noisy speech, including the center front-facing microphone.

*2) Time-Frequency Representation:* The short-time Fourier transform (STFT) coefficients [40] are extracted using a 1024-point Hann window with 75% overlap resulting in $F = 513$. The power spectrogram is the squared absolute of these coefficients.

*3) Model Training:* All models are optimized by RAdam [41], a variant of Adam [42], with a learning rate of $10^{-3}$. Each update is done given a minibatch of 4096 frames, composed of randomly selected 128-frame continuous segments from 32 randomly selected utterances. A voice activity detection is applied to minimize silent frames beforehand. The gradient is normalized with a threshold of 1 [43]. The training of the VAEs and the GF-VAEs begins with a warm-up stage, in which the KL term annealing [44], [45] runs for the first 100 epochs. An early stopping mechanism [46] is used so that all training is stopped after 50 consecutive epochs failed to lower the validation error computed on the development set. The

[2]Demo webpage: https://aanugraha.gitlab.io/demo/taslp20

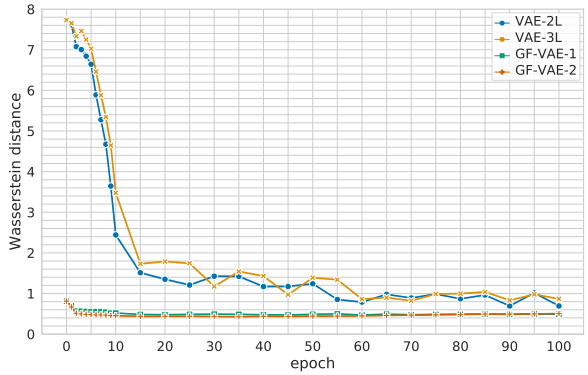



Fig. 5. Wasserstein distance computed between the input and output of the standalone VAEs or the VAE part of the GF-VAEs ($D = 32$) after different training epochs. Lower is better.

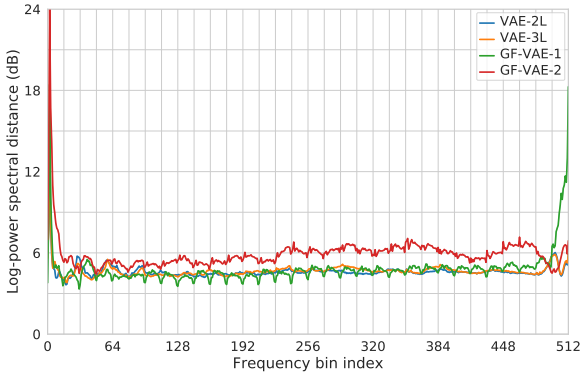



Fig. 6. Log-power spectral distance (LSD) computed on the speech spectrogram reconstructed using the different models ($D = 32$). Lower is better.

model with the lowest validation error is kept and used for the evaluation.

### *B. VAE-Based Model: Transformed vs. Observed Variables*

As stated in Section II-C, we expect that the transformed variables obtained by the GF part of a GF-VAE is easier to model by a VAE than the observed variables. Let $\mathbf{x}_t$ be an observed variable, i.e., a frame of clean speech power spectrum. For the standalone VAE, the input is the observed variable $\mathbf{x}_t$ and the output is the reconstructed observed variable $\widehat{\mathbf{x}}_t \triangleq \boldsymbol{\lambda}_\theta(\mathbf{z}_t)^{-1}$, where $\mathbf{z}_t \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t)$ is the oracle latent variable. For the VAE of a GF-VAE, the input is the transformed variable $\mathbf{y}_t = g_\psi(\mathbf{x}_t)$ and the output is the reconstructed transformed variable $\widehat{\mathbf{y}}_t \triangleq \boldsymbol{\mu}_\theta^{\mathbf{y}}(\mathbf{z}_t)$, where $\mathbf{z}_t \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{y}_t)$ is the oracle latent variable. We gathers the $f$-th element of $\mathbf{x}_t$ for all $T$ frames in a single utterance to construct $\mathbf{x}_f \triangleq \left\{ \{\mathbf{x}_t\}_f \,|\, \forall t \in [1, T] \right\}$. We also construct $\widehat{\mathbf{x}}_f$, $\mathbf{y}_f$, and $\widehat{\mathbf{y}}_f$ in a similar way.

Recall that the VAE part of our GF-VAEs and our VAEs have different assumptions regarding the distribution of their inputs. In a GF-VAE, the input of the VAE part is assumed to have a Gaussian distribution. Conversely, the input of a standalone VAE is assumed to have an exponential distribution. To have a fair comparison, we opt to compute the first Wasserstein distance [47] for each frequency bin of an utterance as $\mathcal{W}_1(\mathbf{x}_f, \widehat{\mathbf{x}}_f)$ for the VAE, or $\mathcal{W}_1(\mathbf{y}_f, \widehat{\mathbf{y}}_f)$ for the VAE of a GF-VAE. The utterance-level distance is then obtained by computing the average across all frequency bins, and Fig. 5 shows the average utterance-level distances computed on the whole test set.

The figure shows that on average, the distribution of $\{\mathbf{x}_f\}_t$ and that of $\{\widehat{\mathbf{x}}_f\}_t$ are getting closer along the training. The figure also shows that the distribution of $\{\mathbf{y}_f\}_t$ and that of $\{\widehat{\mathbf{y}}_f\}_t$ are close, even at the zeroth epoch, i.e., right after the initialization. We observe that at the zeroth epoch, the values of both $\{\mathbf{y}_f\}_t$ and $\{\widehat{\mathbf{y}}_f\}_t$ are close to zero. The close-to-zero values of $\{\mathbf{y}_f\}_t$ can be attributed to the initialization scheme of each flow step (see Section III-B), that makes the activation normalization layer basically does a standardization on each input. We then naturally obtain $\{\widehat{\mathbf{y}}_f\}_t$ that is also close to zero because the initial weights of the VAE are small values around zero. To sum up, this evaluation provides a strong indication that the transformed variable $\mathbf{y}_t$ is easier to model by a VAE than the observed variable $\mathbf{x}_t$.

### *C. Speech Reconstruction Task*

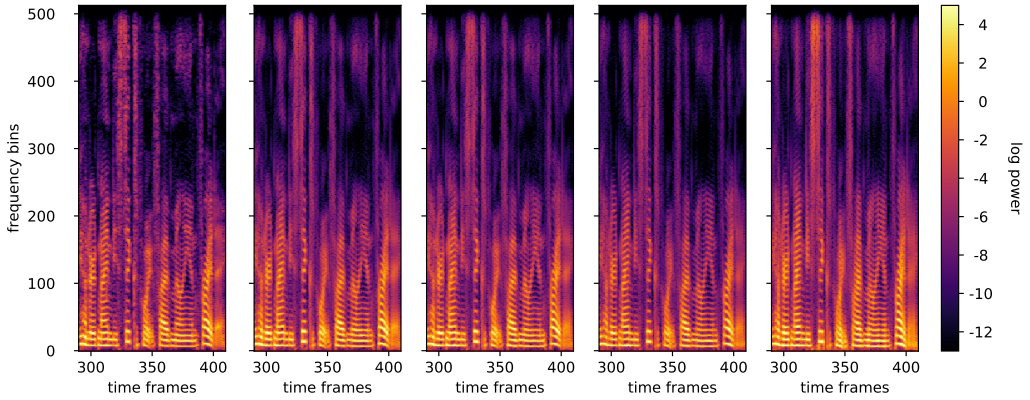
We validate whether a model can reconstruct a speech spectrogram given the oracle latent variables $\mathbf{z} \triangleq [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$ estimated from clean speech $\mathbf{x} \triangleq [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$. The latent variables are obtained using the recognition process, i.e., $\mathbf{z}_t \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(\mathbf{x}_t)$ for the VAE-based speech models and $\mathbf{z}_t \triangleq \boldsymbol{\mu}_\phi^{\mathbf{z}}(g_\psi(\mathbf{x}_t))$ for the GF-VAE-based speech models. The reconstructed spectrogram is then generated by $\widehat{\mathbf{x}}_t \triangleq \boldsymbol{\lambda}_\theta(\mathbf{z}_t)^{-1}$ for the VAE-based models or $\widehat{\mathbf{x}}_t \triangleq g_\psi^{-1}(\boldsymbol{\mu}_\theta^{\mathbf{y}}(\mathbf{z}_t))$ for the GF-VAE-based models. Additionally, we obtain the time-domain speech given the reconstructed spectrogram and the original clean speech phase. Note that the clean speech phase may be inappropriate for the reconstructed, most likely distorted, spectrogram. Nonetheless, it allows us to perform an informal listening test. Since the GF achieves a perfect reconstruction due to its bijective property, we test only the VAEs and the GF-VAEs.

To have an objective measure on the reconstructed power spectrogram, we compute the log-power spectral distance (LSD) [48] for each TF bin and present the average over all frames of the test set in Fig. 6. The LSD is computed as

$$\mathrm{LSD}_{ft} = 10 \left| \log_{10} \left( \{\mathbf{x}_t\}_f \right) - \log_{10} \left( \{\widehat{\mathbf{x}}_t\}_f \right) \right|, \qquad (32)$$

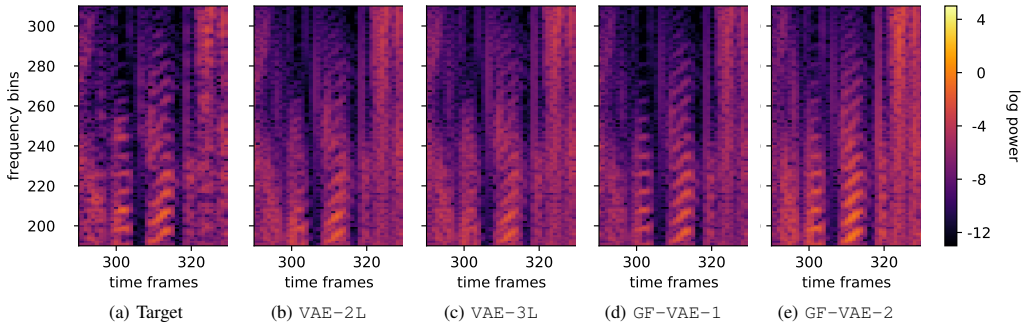
where $|\cdot|$ returns the absolute value.

We also do a visual assessment on the reconstructed spectrogram. As shown in Fig. 7, all models are able to reproduce the lower part of the speech harmonics. Above $f \geq 200$ ($\approx$ 3.1 kHz), the VAEs' harmonics start to be blurry. By contrast, as exhibited in Fig. 8, the GF-VAEs keep preserving the harmonics for those higher frequency parts. Although we have observed that the GF-VAEs preserve the harmonics better than the VAEs, Fig. 6 shows that the average LSDs of `GF-VAE-1` and the VAEs are close in most cases and are better than those of `GF-VAE-2`. For `GF-VAE-1`, we can easily observe that there are periodic downward spikes happened every 16 frequency bins. It coincides with the hyperparameter of the *squeeze* function, which splits the spectrogram along the




(a) Target (b) VAE-2L (c) VAE-3L (d) GF-VAE-1 (e) GF-VAE-2

Fig. 7. Log-power spectrogram examples of the speech reconstructed using the different models ($D = 32$). The segments are from the utterance F05_442C020T_PED from the test set et05_ped_simu.




(a) Target (b) VAE-2L (c) VAE-3L (d) GF-VAE-1 (e) GF-VAE-2

Fig. 8. Harmonic structure examples of the speech reconstructed using the different models ($D = 32$). These examples are parts of the log-power spectrograms shown in Fig. 7. The segments are from the utterance F05_442C020T_PED from the test set et05_ped_simu.

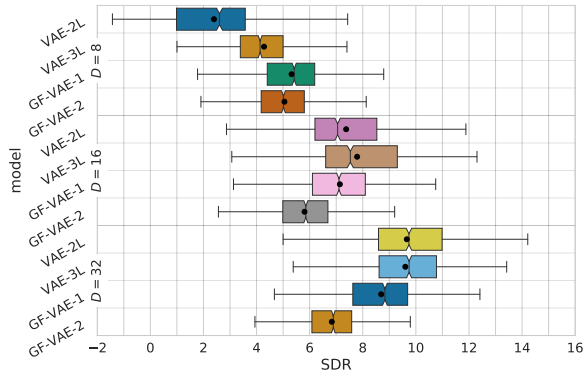



Fig. 9. Boxplots of signal distortion ratio (SDR) metric computed on the speech reconstructed using the different models. The black dots are the mean values. Higher is better.

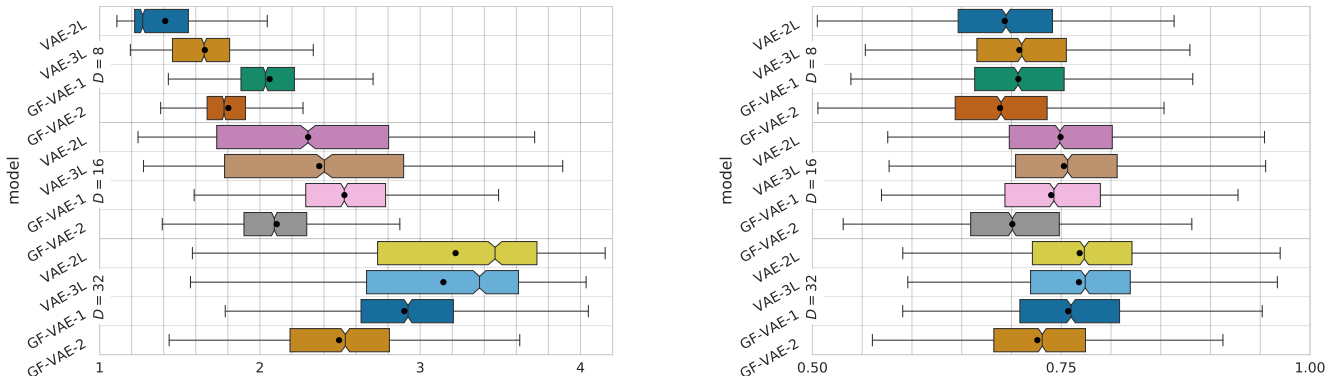
frequency axis into 16 feature maps (see Section III-B). There may be a trade-off between the use of a single *squeeze* function, as in this paper and [29], and the use of multiple *squeeze* functions, as in [20]. We leave further investigation on the GF design for future work.

We then evaluate the reconstructed time-domain speech in terms of the signal-to-distortion ratio (SDR) score [49], the wideband extension of the Perceptual Evaluation of Speech Quality (WB-PESQ) score [50], [51], and the Short-Time Objective Intelligibility (STOI) score [52]. Interestingly, our visual assessment above is not reflected on these objective metrics as shown in Figs. 9 and 10. In terms of SDR, the GF-VAEs are better than the VAEs for $D = 8$, and the opposite for $D = 32$. In terms of WB-PESQ and STOI, GF-VAE-1 is generally better than the VAEs for $D \in \{8, 16\}$, and the VAEs are better than the GF-VAEs for $D = 32$. Nonetheless, the performance of the GF-VAEs has smaller interquartile ranges in most cases, especially for the SDR and WB-PESQ scores, and thus, is more stable than that of the VAEs.

To conclude, although we observe that the GF-VAEs preserve the harmonics better than the VAEs, all of our VAEs and GF-VAEs can reconstruct a satisfying speech power spectrogram given the oracle latent variables. To obtain a time-domain signal given a reconstructed power spectrogram, we use the clean phase, which is a simple, but sub-optimal, way. Several works [53]–[55] propose better methods in estimating the phase given a spectrogram. This kind of phase estimation methods should be useful if we want to have a good time-domain signal.

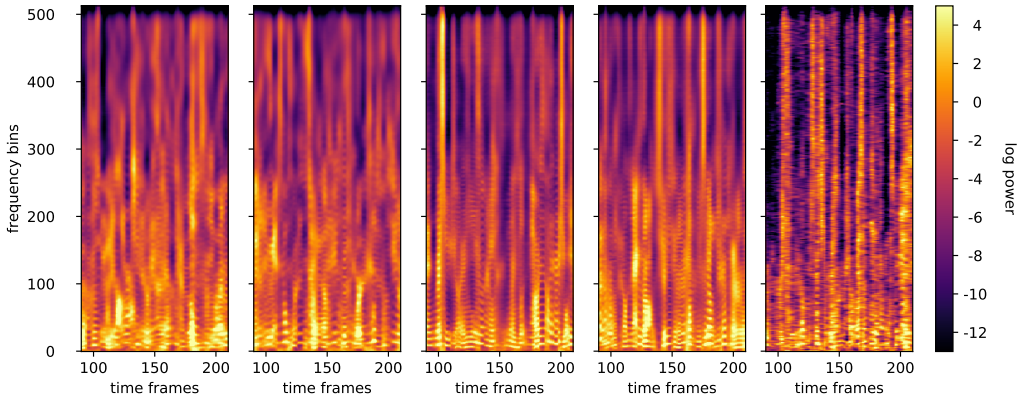
### D. Random Speech Generation Task

We have observed that our VAEs and GF-VAEs can reconstruct a satisfying speech power spectrogram. We now validate whether a model can generate a speech spectrogram, that is characterized by its harmonic structures, given random latent variables $\mathbf{z} \triangleq [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$ or transformed variables $\mathbf{y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$. For this, we first randomly sample 65




(a) Wide-band Perceptual Evaluation of Speech Quality (WB-PESQ) (b) Short-Time Objective Intelligibility (STOI)

Fig. 10. Boxplots of perceptual objective metrics computed on the speech reconstructed using the different models. The black dots are the mean values. Higher is better.




(a) VAE-2L (b) VAE-3L (c) GF-VAE-1 (d) GF-VAE-2 (e) GF

Fig. 11. Log-power spectrogram examples of the speech generated from random latent variables ($D = 32$) or transformed variables using the different models.

vectors from $p(\mathbf{z}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ or $p(\mathbf{y}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then perform interpolation so that we obtain a slowly-evolving sequence of 256 vectors as $\mathbf{z}$ or $\mathbf{y}$.

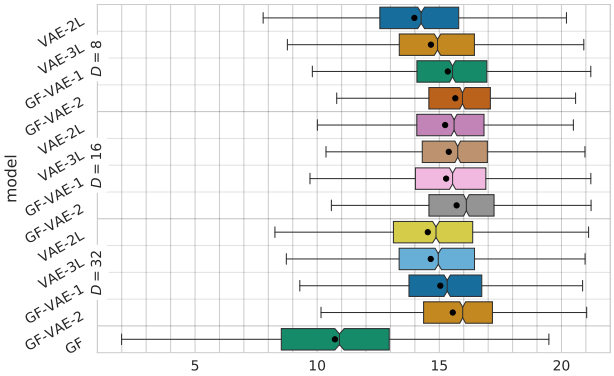
As shown in Fig. 11, the VAEs and the GF-VAEs can generate speech-like harmonic structures given some random latent variables. The GF-VAEs produce finer structures than the VAEs and, in general, the VAEs' output tends to be smoother than the GF-VAEs'. By contrast, the GF's output does not resemble a speech spectrogram because there is no noticeable harmonic structure. This suggests that the GF lacks a generalization capability and thus, it might not be suitable for providing the speech prior for the speech enhancement task below.

### *E. Multichannel Speech Enhancement Task*

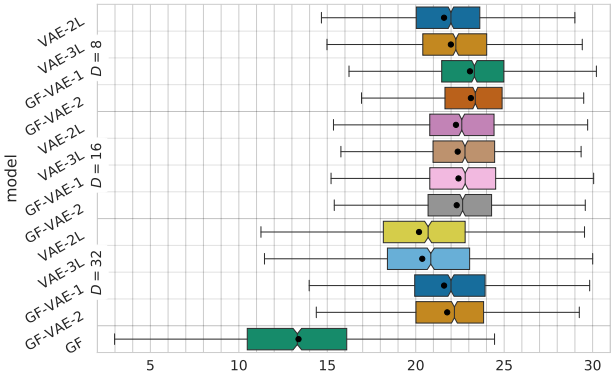
We use each model as the speech spectrogram model (a.k.a. the deep speech prior) for the semi-supervised multichannel speech enhancement method [7] described in Section IV. The use of a deep speech model (either a GF, a VAE, or a GF-VAE) as the deep speech prior is detailed in Section IV-A. Additionally, the noise spectrogram model is provided by an NMF whose number of basis vectors is $L = 32$. The number of channels is $M = 5$. The number of parameter update iterations is 128 and in each iteration, the latent variables are updated by the Metropolis sampling method [10] for 32 times.

The performance is evaluated using the BSS-Eval toolbox [56] to compute the signal-to-distortion ratio (SDR), the signal-to-interferences ratio (SIR), and the signal-to-artifacts ratio (SAR) on the enhanced 5-channel speech as shown in Fig. 12. As expected from the speech random generation evaluation above, the GF-VAEs and the VAEs clearly outperform the GF. The SDRs, that are regarded as the overall performance metrics, show that the GF-VAEs significantly outperform the VAEs in most cases and the GF-VAE-2 always outperforms the GF-VAE-1 and the VAEs. The VAE-3L significantly outperforms the VAE-2L only for $D = 8$. It indicates that increasing the number of layers alone is not effective to achieve a performance improvement. Most importantly, the GF-VAEs are more robust to the setting of latent variable dimension than the VAEs. The SDR medians for the different latent variable dimensions $D = 8$, $D = 16$, and $D = 32$ are 14.3, 15.6, and 14.9, respectively, for the VAE-2L, and 14.9, 15.8, and 15.0 for the VAE-3L. By contrast, those are 15.5, 15.5, and 15.3 for the GF-VAE-1, and 15.9, 16.1, and 16.0 for the GF-VAE-2. Interestingly, the SIRs indicate that the noise removal capability of both GF-VAEs is not significantly different. Thus, the overall performance difference between two GF-VAEs can be attributed to the difference of artifacts as shown by the SARs.

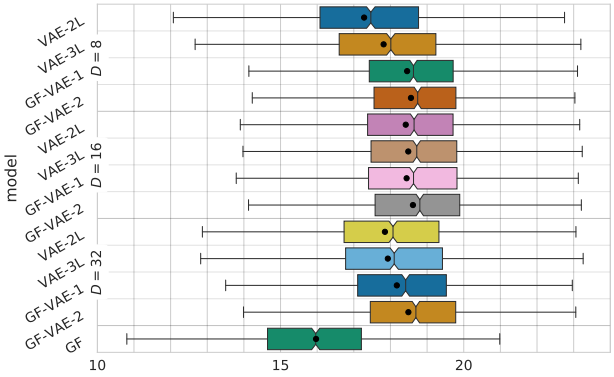
In addition to measuring the performance on the multichannel enhanced speech, we also evaluate the performance on a




(a) Signal to Distortion Ratio (SDR)




(b) Signal to Interferences Ratio (SIR)




(c) Signal to Artifacts Ratio (SAR)

Fig. 12. Boxplots of source separation metrics computed on the multichannel speech enhanced using the different models. The black dots are the mean values. Higher is better.

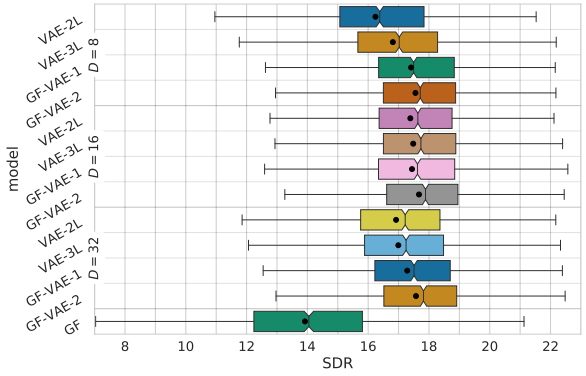
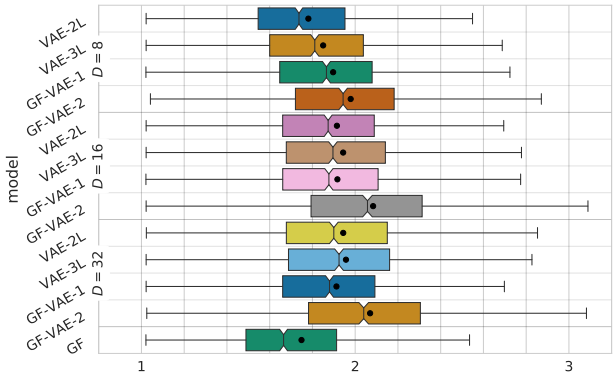
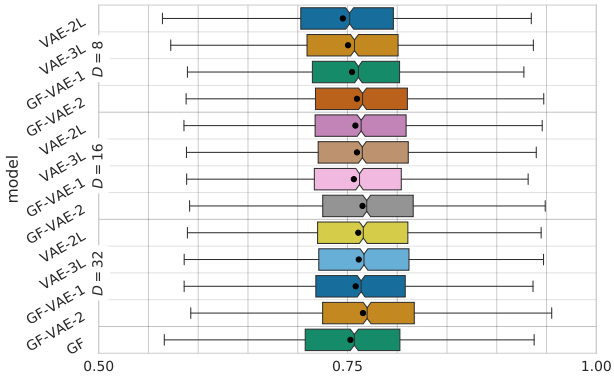



Fig. 13. Boxplots of signal distortion ratio (SDR) metric computed on a single channel, corresponding to the center front-facing microphone, of the multichannel speech enhanced using the different models. The black dots are the mean values. Higher is better.




(a) Wide-band Perceptual Evaluation of Speech Quality (WB-PESQ)

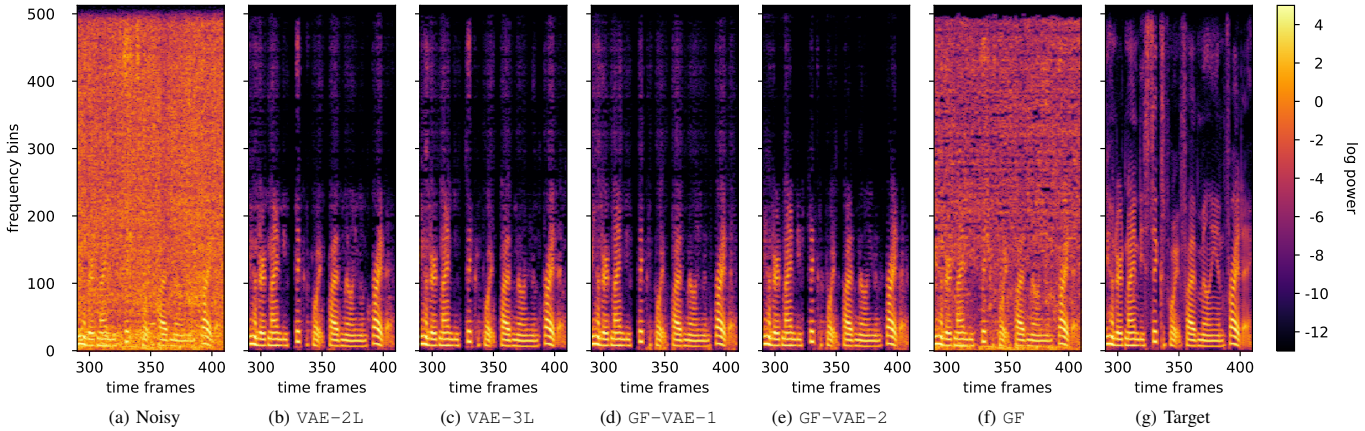



(b) Short-Time Objective Intelligibility (STOI)

Fig. 14. Boxplots of perceptual objective metrics computed on a single channel, corresponding to the center front-facing microphone, of the multichannel speech enhanced using the different models. The black dots are the mean values. Higher is better.

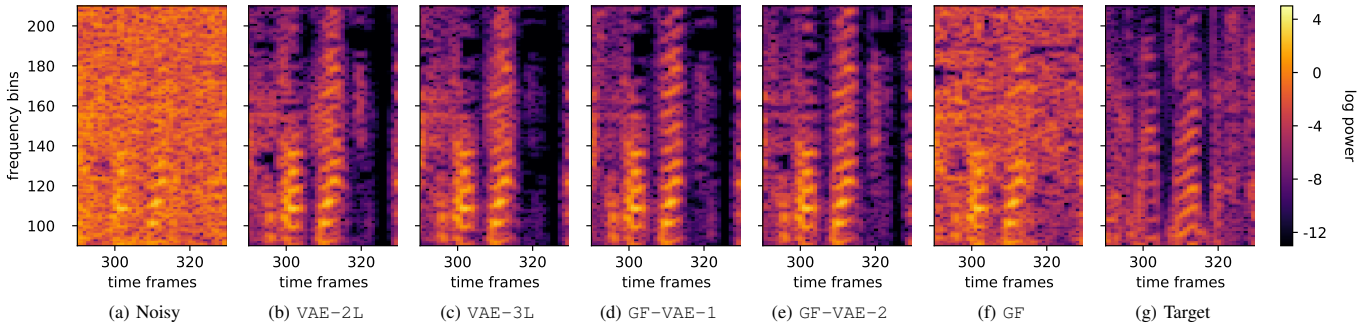
single channel (the center front-facing microphone) of the multichannel enhanced speech. We consider the SDR [49], the WB-PESQ [50], [51], and the STOI [52] as shown in Figs. 13 and 14. As computed on the multichannel enhanced speech, in terms of SDR, the GF-VAEs significantly outperform the VAEs in most cases and the `GF-VAE-2` always significantly outperforms the `GF-VAE-1` and the VAEs. The `GF-VAE-2` also always significantly outperforms the other models in terms of WB-PESQ. The differences of STOI scores are not statistically significant in most cases, but we can still observe that the `GF-VAE-2` tends to have higher score than the other models. In short, the `GF-VAE-2` provides the lowest signal distortion, indicated by the SDR score, and the lowest perceptual distortion, indicated by the WB-PESQ and STOI scores.

Examples of speech spectrograms enhanced using the different models are shown in Fig. 15. While the noise is minimally reduced using the `GF`, we can effectively separate the speech from the noise using the VAEs or the GF-VAEs. As we already observed in the speech reconstruction and the speech generation tasks, the GF-VAEs produce better, although subtle, fine structures than the VAEs as demonstrated in Fig. 16. The spectrograms shown in Figs. 15 and 16 are the ones of




(a) Noisy (b) VAE-2L (c) VAE-3L (d) GF-VAE-1 (e) GF-VAE-2 (f) GF (g) Target

Fig. 15. Log-power spectrogram examples of the speech enhanced using the different models compared to the input noisy spectrogram and the target clean spectrogram. The latent variable dimension for the VAEs and the GF-VAEs is $D = 16$. The segments are from the utterance F05_442C020T_PED from the test set et05_ped_simu.




(a) Noisy (b) VAE-2L (c) VAE-3L (d) GF-VAE-1 (e) GF-VAE-2 (f) GF (g) Target

Fig. 16. Harmonic structure examples of the speech enhanced using the different models compared to the input noisy spectrogram and the target clean spectrogram. These examples are parts of the log-power spectrograms shown in Fig. 15. The latent variable dimension for the VAEs and the GF-VAEs is $D = 16$. The segments are from the utterance F05_442C020T_PED from the test set et05_ped_simu.

the center front-facing microphone.

In conclusion, the GF-VAE-2 is the best choice for the considered speech enhancement task. Most importantly, the GF should not be used to provide the speech prior for a statistical semi-supervised speech enhancement method like the one we use in this task.

### F. Discussion

An interesting observation is that while the GF-VAEs tend to underperform the VAEs in the speech reconstruction task (Section V-C), the GF-VAEs tend to outperform the VAEs in the speech enhancement task (Section V-E). Note that the absolute SDR, WB-PESQ, and STOI scores cannot be compared directly between these tasks because different data are processed with different method complexities. The reconstruction task processes a single-channel clean speech spectrogram. The latent variables are simply estimated using a recognition model and then used, without any further optimization, for the reconstruction using a generative model. The time-domain speech signal is obtained given the reconstructed spectrogram and the original clean speech phase, which may not be suitable for the reconstructed spectrogram. Conversely, the enhancement task processes a multichannel noisy speech spectrogram. Although the estimated latent variables are also initialized using a recognition model, they are updated multiple times so that the generated spectrogram maximizes the objective function. The time-domain speech signal with appropriate phase is obtained by multichannel Wiener filtering given the generated speech spectrogram and other parameters, including the noise spectrogram, the speech spectrogram scaling factors, and the spatial parameters.

## VI. Conclusion

This paper proposes a deep generative model called the GF-VAE for modeling high-dimensional observed variables by utilizing a variational autoencoder (VAE) to discover low-dimensional latent variables from high-dimensional transformed variables obtained by a generative flow (GF). Through evaluation in the context of speech power spectrogram modeling, we showed that the GF-VAE can reconstruct the fine harmonics in the higher frequency bands better than the VAE. We also showed that the GF-VAE outperformed the others for a semi-supervised multichannel speech enhancement.

Future directions include incorporating temporal dependency structures and non-Gaussian latent variable spaces [23], [57] into the GF-VAE framework, and extending GF-VAE to deal with complex-valued spectrograms with phase information [58]. Further, in a similar way that the VAE is used with conditional inputs [59], a conditional GF-VAE would be useful for other

applications, including source separation [60], speech synthesis [61], voice conversion [62], and music composition [63].

## REFERENCES


[1] E. Vincent, T. Virtanen, and S. Gannot, Eds., *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[2] S. Makino, Ed., *Audio Source Separation*. Springer, 2018.

[3] 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 ICASSP*, Calgary, Canada, 2018, pp. 716–720.

[4] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE MLSP*, Aalborg, Denmark, 2018, pp. 1–6.

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

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

[7] 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, Language Process.*, vol. 27, no. 12, pp. 2197–2212, 2019.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, Banff, Canada, 2014, pp. 1–9.

[9] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer, 2010.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[11] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 04 1970.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[13] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Semi-blind source separation with multichannel variational autoencoder," pp. 1–8, 2018, arXiv:1808.00892v3.

[14] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 3158–3162.

[15] O. Bousquet, S. Gelly, I. Tolstikhin, C.-J. Simon-Gabriel, and B. Schoelkopf, "From optimal transport to generative modeling: the VEGAN cookbook," pp. 1–11, 2017, arXiv:1705.07642v1.

[16] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *Proc. ICLR*, Vancouver, Canada, 2018, pp. 1–11.

[17] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," in *Proc. ICLR*, New Orleans, USA, 2019, pp. 1–12.

[18] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, Lille, France, 2015, pp. 1530–1538.

[19] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proc. ICLR*, Toulon, France, 2017, pp. 1–12.

[20] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Proc. NIPS*, Montréal, Canada, 2018, pp. 10 215–10 224.

[21] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[22] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[23] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," in *Proc. UAI*, Monterey, USA, 2018, pp. 856–865.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[25] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *Proc. ICLR*, San Diego, USA, 2015, pp. 1–11.

[26] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving variational inference with inverse autoregressive flow," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 4743–4751.

[27] H. P. Das, P. Abbeel, and C. J. Spanos, "Dimensionality reduction flows," pp. 1–10, 2019, arXiv:1908.01686v1.

[28] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," pp. 1–12, 2016, arXiv:1609.03499v2.

[29] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE ICASSP*, Brighton, UK, 2019, pp. 3617–3621.

[30] S. Kim, S. Lee, J. Song, and S. Yoon, "FloWaveNet : A generative flow for raw audio," in *Proc. ICML*, Long Beach, USA, 2019, pp. 3370–3378.

[31] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, 2015, pp. 448–456.

[33] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 7, pp. 1830–1840, Sep. 2010.

[34] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[35] K. Lange, *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics, 2016.

[36] T. Ando, C.-K. Li, and R. Mathias, "Geometric means," *Linear Algebra and its Applications*, vol. 385, pp. 305–334, 2004.

[37] W.-H. Chen, "A review of geometric mean of positive definite matrices," *British Journal of Mathematics & Computer Science*, vol. 5, no. 1, pp. 1–12, 2015.

[38] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE ASRU*, Scottsdale, USA, Dec. 2015, pp. 504–511.

[39] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[40] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 3, pp. 235–238, 1977.

[41] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. ICLR*, Addis Ababa, Ethiopia, 2020, pp. 1–10.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, USA, 2015, pp. 1–11.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, Atlanta, USA, 2013, pp. 1310–1318.

[44] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. CoNLL*, Berlin, Germany, 2016, pp. 10–21.

[45] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in *Proc. NIPS*, 2016, pp. 3738–3746.

[46] L. Prechelt, "Early stopping – but when?" in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer, 2012, pp. 53–67.

[47] A. Ramdas, N. Trillos, and M. Cuturi, "On Wasserstein two-sample testing and related families of nonparametric tests," *Entropy*, vol. 19, no. 2, pp. 1–15, 2017.

[48] L. Rabiner, L. Rabiner, and B. Juang, *Fundamentals of Speech Recognition*. PTR Prentice Hall, 1993.

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

[50] *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, ITU-T Recommendation P.862, 2001.

[51] *Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs*, ITU-T Recommendation P.862.2, 2007.

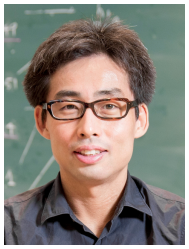
[52] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[53] S. Takamichi, Y. Saito, N. Takamune, D. Kitamura, and H. Saruwatari, "Phase reconstruction from amplitude spectrograms based on von-Mises-distribution deep neural network," in *Proc. IWAENC*, Tokyo, Japan, 2018, pp. 286–290.

[54] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in *Proc. EUSIPCO*, Rome, Italy, 2018, pp. 2514–2518.
[55] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep Griffin-Lim iteration," in *Proc. IEEE ICASSP*, Brighton, UK, 2019, pp. 61–65.
[56] E. Vincent, H. Sawada, P. Bofill, S. Makino, and J. P. Rosca, "First stereo audio source separation evaluation campaign: Data, algorithms and results," in *Proc. ICA*, 2007, pp. 552–559.
[57] J. Xu and G. Durrett, "Spherical latent spaces for stable variational autoencoders," in *Proc. EMNLP*, Brussels, Belgium, 2018, pp. 4503–4513.
[58] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A deep generative model of speech complex spectrograms," in *Proc. IEEE ICASSP*, Brighton, UK, 2019, pp. 905–909.
[59] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. NIPS*, Montréal, Canada, 2015, pp. 3483–3491.
[60] 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. IEEE ICASSP*, Brighton, UK, 2019, pp. 546–550.
[61] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," in *Proc. ICLR*, New Orleans, USA, 2019, pp. 1–13.
[62] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. IEEE ICASSP*, Calgary, Canada, 2018, pp. 5274–5278.
[63] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proc. ISMIR*, Paris, France, 2018, pp. 747–754.

**Aditya Arie Nugraha** received the B.S. and M.S. degrees in electrical engineering from Institut Teknologi Bandung, Indonesia, in 2008 and 2011, respectively, the M.E. degree in computer science and engineering from Toyohashi University of Technology, Japan, in 2013, and the Ph.D. degree in informatics from Université de Lorraine and Inria Nancy–Grand-Est, France, in 2017. He was a research engineer at Inria Nancy–Grand-Est in 2018. He is currently a Postdoctoral Researcher at the Center for Advanced Intelligence Project (AIP), RIKEN, Japan. His research interests include audio signal processing and machine learning.

**Kouhei Sekiguchi** received the B.E. and M.S. degrees from Kyoto University, Kyoto, Japan, in 2015 and 2017, respectively. He is currently a researcher at the Center for Advanced Intelligence Project (AIP), RIKEN, Japan, and working toward the Ph.D. degree in Kyoto University. His research interests include microphone array signal processing and machine learning. He is a member of IEEE and IPSJ.



**Kazuyoshi Yoshii** received the M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2005 and 2008, respectively. He is an Associate Professor at the Graduate School of Informatics, Kyoto University, and concurrently the Leader of the Sound Scene Understanding Team, Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo, Japan. His research interests include music informatics, audio signal processing, and statistical machine learning.