

# A BAYESIAN PERMUTATION TRAINING DEEP REPRESENTATION LEARNING METHOD FOR SPEECH ENHANCEMENT WITH VARIATIONAL AUTOENCODER

Yang Xiang<sup>\*†</sup>, Jesper Lisby Højvang<sup>†</sup>, Morten Højfeldt Rasmussen<sup>†</sup>, Mads Græsbøll Christensen<sup>\*</sup>

<sup>\*</sup> Audio Analysis Lab, CREATE, Aalborg University, Aalborg, Denmark {yaxi,mgc}@create.aau.dk

<sup>†</sup> Capturi A/S, Aarhus, Denmark {yxi,jlh,mhr}@capturi.com

## ABSTRACT

Recently, variational autoencoder (VAE), a deep representation learning (DRL) model, has been used to perform speech enhancement (SE). However, to the best of our knowledge, current VAE-based SE methods only apply VAE to model speech signal, while noise is modeled using the traditional non-negative matrix factorization (NMF) model. One of the most important reasons for using NMF is that these VAE-based methods cannot disentangle the speech and noise latent variables from the observed signal. Based on Bayesian theory, this paper derives a novel variational lower bound for VAE, which ensures that VAE can be trained in supervision, and can disentangle speech and noise latent variables from the observed signal. This means that the proposed method can apply the VAE to model both speech and noise signals, which is totally different from the previous VAE-based SE works. More specifically, the proposed DRL method can learn to impose speech and noise signal priors to different sets of latent variables for SE. The experimental results show that the proposed method can not only disentangle speech and noise latent variables from the observed signal, but also obtain a higher scale-invariant signal-to-distortion ratio and speech quality score than the similar deep neural network-based (DNN) SE method.

**Index Terms**— Deep representation learning, speech enhancement, Bayesian permutation training, variational autoencoder

## 1. INTRODUCTION

In real-world environments, speech signals are often distorted due to the presence of background noise. To reduce the effects of noise, speech enhancement (SE) techniques have been developed [1, 2] to improve the quality and intelligibility of an observed signal.

Currently, many single-channel SE algorithms have been proposed, which include some unsupervised algorithms [3, 4] and supervised algorithms [5, 6]. However, these methods usually apply linear processes to model complex high-dimensional signal, which is not always reasonable in practical applications [7]. Thus, non-linear deep neural network (DNN) models have been developed. As shown in [2, 7–10], DNN-based methods can achieve better SE performance than traditional linear models. However, their generalization ability is not often satisfactory for the unseen noise conditions [2].

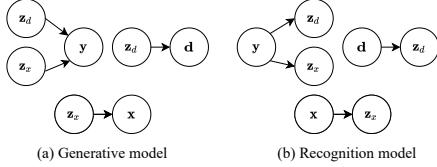
Recently, deep probabilistic generative models have been investigated to improve the DNN’s generalization ability for SE, such as generative adversarial networks (GAN) [11] and the variational autoencoder (VAE) [12, 13]. VAE can learn the probability distribution of complex data and perform efficient approximate posterior inference, so VAE-based SE algorithms have been proposed [13–15].

This work was partly supported by Innovation Fund Denmark (Grant No.9065-00046).

However, the VAE of these methods is trained in an unsupervised manner on speech only, and the noise is modeled by an NMF model because these methods cannot disentangle the speech and noise latent variables from the observed signal. This means that these algorithms are not robust [16], and their SE performance is limited compared to DNN-based supervised methods [13]. To mitigate this problem, supervised VAE-based SE methods have been proposed. In [16, 17], a supervised classifier [16] and a supervised noise-aware training strategy [17] are introduced to the training of speech VAE. The purpose is to obtain a more robust speech latent variable from the observed signal. However, the noise is still modeled by a linear NMF model because it is a difficult task to disentangle the speech and noise latent variables from the observed signal [17].

Learning interpretable latent representation is a challenging but very useful task because it can explain how different factors influence the speech signal, which is important in speech-related applications [18]. In [18], a latent space arithmetic operation was derived to modify the speech attributes (phonetic content and speaker identity). [19] proposed an unsupervised method to distinguish different latent variables and generate new latent variables for the ASR application. [20] applied VAE to learn the sequence-dependent and sequence-independent representations. However, interpretable latent representation is rarely considered in current SE algorithms [7–10].

Inspired by previous work, in this paper, we propose a Bayesian permutation training method for SE. The proposed method can disentangle the latent speech and noise variables from the observed signal in a supervised manner and conduct the mapping between latent variables and targets, even though this is a difficult task [17]. We hypothesize that disentangling latent variables can improve the performance of the supervised DNN-based SE method. To achieve this, a clean speech VAE (C-VAE) and a noise VAE (N-VAE) are separately pre-trained without supervision. After that, based on Bayesian theory and our derived variational lower bound, we use the two pre-trained VAEs to train a noisy VAE (NS-VAE) in a supervised manner. The trained NS-VAE can learn the latent representations of the speech and noise signal. When we conduct SE, the trained NS-VAE is first used to predict the latent variables of the speech and noise signal. Then, the two latent variables are independently used as the decoder input of the C-VAE and N-VAE to estimate the corresponding speech and noise. Finally, the enhanced signal can be acquired by direct speech waveform reconstruction or with post-filtering methods. Compared to previous VAE-based SE methods [13–17] and interpretable latent representation learning methods [18–20], the proposed method derives a novel variational lower bound to ensure that supervision training can be used for VAE, and VAE can disentangle different latent variables to model noise for SE. The derived supervised lower bound is very different from previous VAE-based methods [12–20] that are trained on an unsupervised variational lower bound, which increases the robustness of different learned latent



**Fig. 1:** Graphical illustration of the proposed model.

variables [17]. Moreover, our learned latent variables are attributed to different types of signal, so each single latent variable that is generated by NS-VAE can be used to generate the corresponding speech or noise signal, and their combination can generate noisy speech.

## 2. PROBLEM DESCRIPTION

In this work, we aim to perform SE in an additive noisy environment. Thus, the signal model can be written as

$$y(t) = x(t) + d(t), \quad (1)$$

where  $y(t)$ ,  $x(t)$ , and  $d(t)$  represent the observed, speech, and noise signal, respectively, and  $t$  is the time index. Log-power spectrum (LPS) is suitable for direct signal estimation [7], so we use it as a feature for SE. The LPS of  $y(t)$ ,  $x(t)$ , and  $d(t)$  is written as  $Y(f, n)$ ,  $X(f, n)$ , and  $D(f, n)$ , respectively. Here,  $f \in [1, F]$  and  $n \in [1, N]$  denote the frequency bins and time frame indices, respectively. Collecting  $F$  frequency bins and  $N$  time frames, we can obtain the LPS dataset  $\mathbf{Y}_N$ ,  $\mathbf{X}_N$  and  $\mathbf{D}_N$  with  $N$  samples, where  $\mathbf{Y}_N = [y_1, \dots, y_n, \dots, y_N]$  and  $y_n = [Y(1, n), \dots, Y(f, n), \dots, Y(F, n)]^T$ ,  $\mathbf{x}_n$  and  $\mathbf{d}_n$  are defined similarly to  $\mathbf{y}_n$ .  $\mathbf{X}_N$  and  $\mathbf{D}_N$  are defined similarly to  $\mathbf{Y}_N$ . For simplicity, we use  $\mathbf{y}$ ,  $\mathbf{x}$ , and  $\mathbf{d}$  to represent a sample in dataset  $\mathbf{Y}_N$ ,  $\mathbf{X}_N$  and  $\mathbf{D}_N$ , respectively. In the proposed VAE model, we assume that  $\mathbf{y}$  is generated from a random process involving the speech latent variables  $\mathbf{z}_x$  and the noise latent variables  $\mathbf{z}_d$ , where the observed speech conditional prior distribution can be written as  $q(\mathbf{y}|\mathbf{z}_d, \mathbf{z}_x)$ . The dimensions of vectors  $\mathbf{z}_x$  and  $\mathbf{z}_d$  are  $L_x$  and  $L_d$ , respectively. The dataset of  $\mathbf{z}_x$  and  $\mathbf{z}_d$  is written as  $\mathbf{Z}_{xN}$  and  $\mathbf{Z}_{dN}$  with  $N$  samples. Here, we assume that the latent variables  $\mathbf{z}_x$  and  $\mathbf{z}_d$  are independent. Additionally,  $\mathbf{x}$  and  $\mathbf{d}$  are drawn from the speech prior distribution  $q(\mathbf{x}|\mathbf{z}_x)$  and the noise prior distribution  $q(\mathbf{d}|\mathbf{z}_d)$ , respectively. The whole generative process is illustrated in Fig. 1(a). In VAE, since exact posterior inference is intractable, we propose that  $\mathbf{z}_x$  and  $\mathbf{z}_d$  can be estimated from speech and noise posterior distributions  $p(\mathbf{z}_x|\mathbf{x})$  and  $p(\mathbf{z}_d|\mathbf{d})$ , respectively, or they can also be estimated from the noisy speech posterior distributions  $p(\mathbf{z}_x|\mathbf{y})$  and  $p(\mathbf{z}_d|\mathbf{y})$ . Here, we assume  $p(\mathbf{z}_x, \mathbf{z}_d|\mathbf{y}) = p(\mathbf{z}_x|\mathbf{y})p(\mathbf{z}_d|\mathbf{y})$ , which ensures that noise can be modeled by non-linear VAE rather than NMF. Based on these assumptions and our derivation in section 3, speech and noise latent variables can be obtained from the observed signal. The whole recognition process is shown in Fig. 1(b).

To sum up, we intend to first estimate the latent variable distributions of the speech  $p(\mathbf{z}_x|\mathbf{y})$  and the noise  $p(\mathbf{z}_d|\mathbf{y})$  from the observed signal to acquire latent variables  $\mathbf{z}_x$  and  $\mathbf{z}_d$ , respectively. After that, we use the estimated latent variables as the input of the decoder of C-VAE and N-VAE to obtain the probability distribution of  $q(\mathbf{x}|\mathbf{z}_x)$  and  $q(\mathbf{d}|\mathbf{z}_d)$  for SE.

## 3. SE WITH BAYESIAN PERMUTATION TRAINING

### 3.1. Variational autoencoder with multiple latent variables

VAE [12] defines a probabilistic generative process between observed signal and its latent variables and provides a principled method to jointly learn latent variables, generative and recognition models, which is achieved by maximizing variational lower bound using stochastic gradient descent (SGD) algorithm. This optimizing process [12] is equal to minimize Kullback-Leibler (KL) divergence

$(D_{KL})$  between real joint probability distribution  $p(\mathbf{y}, \mathbf{z}_x, \mathbf{z}_d)$  and its estimation  $q(\mathbf{y}, \mathbf{z}_x, \mathbf{z}_d)$ . This process can be written as follows:

$$\begin{aligned} D_{KL}(p(\mathbf{y}, \mathbf{z}_x, \mathbf{z}_d) || q(\mathbf{y}, \mathbf{z}_x, \mathbf{z}_d)) &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\log p(\mathbf{y})] \\ &+ \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_x, \mathbf{z}_d|\mathbf{y})) || q(\mathbf{z}_x, \mathbf{z}_d)]. \end{aligned} \quad (2)$$

In (2), the term  $\mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\log p(\mathbf{y})]$  is a constant, so minimizing their KL divergence is equal to minimizing

$$\begin{aligned} \mathcal{L}(\theta_y, \varphi_y, \theta_x, \varphi_x, \theta_d, \varphi_d; \mathbf{y}) &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_x, \mathbf{z}_d|\mathbf{y})) || q(\mathbf{z}_x, \mathbf{z}_d))] \\ &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_x, \mathbf{z}_d|\mathbf{y})) || q(\mathbf{z}_x, \mathbf{z}_d))] \\ &\quad - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\mathbb{E}_{\mathbf{z}_d, \mathbf{z}_x \sim p(\mathbf{z}_d, \mathbf{z}_x|\mathbf{y})} [\log q(\mathbf{y}|\mathbf{z}_x, \mathbf{z}_d)]], \end{aligned} \quad (3)$$

where  $\theta_y, \varphi_y, \theta_x, \varphi_x, \theta_d, \varphi_d$  are the parameters that are used to conduct the related probability estimation. The details will be presented later. Here,  $-\mathcal{L}$  can be seen as the VAE variational lower bound with multiple latent variables ( $\mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\log q(\mathbf{y})] \geq -\mathcal{L}$ ) [12]. Minimizing  $\mathcal{L}$  is equal to maximize this variational lower bound. Based on our assumptions in section 2 ( $\mathbf{z}_x$  and  $\mathbf{z}_d$  are independent and  $p(\mathbf{z}_x, \mathbf{z}_d|\mathbf{y}) = p(\mathbf{z}_x|\mathbf{y})p(\mathbf{z}_d|\mathbf{y})$ ), (3) can be further written as

$$\begin{aligned} \mathcal{L}(\theta_y, \varphi_y, \theta_x, \varphi_x, \theta_d, \varphi_d; \mathbf{y}) &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_x|\mathbf{y}) || q(\mathbf{z}_x))] \\ &+ \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_d|\mathbf{y}) || q(\mathbf{z}_d))] \\ &\quad - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\mathbb{E}_{\mathbf{z}_d, \mathbf{z}_x \sim p(\mathbf{z}_d, \mathbf{z}_x|\mathbf{y})} [\log q(\mathbf{y}|\mathbf{z}_x, \mathbf{z}_d)]], \end{aligned} \quad (4)$$

### 3.2. DRL with Bayesian permutation training

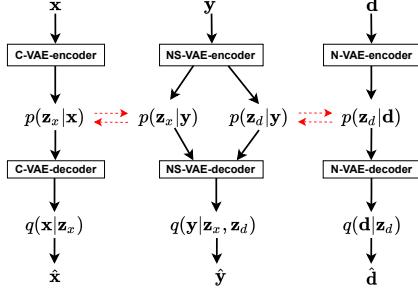
To estimate the speech and noise latent variables from the observed signal using (4), we propose a Bayesian permutation training process between NS-VAE, C-VAE, and N-VAE. First, the C-VAE and N-VAE are separately pre-trained using the general VAE training method [12] without supervision. The purpose is to acquire the posterior estimates  $p(\mathbf{z}_x|\mathbf{x})$  and  $p(\mathbf{z}_d|\mathbf{d})$ . Then, the NS-VAE is trained with the supervision of C-VAE and N-VAE.

In (4), the calculation of the first and second term is similar, so we will only use the first term to show the Bayesian permutation process. To achieve supervision learning, we add an attention mechanism ( $p(\mathbf{z}_x|\mathbf{x})/p(\mathbf{z}_x|\mathbf{x})$ ) for the calculation of first term in (4). Thus, its calculation can be written as (5)

$$\begin{aligned} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [D_{KL}(p(\mathbf{z}_x|\mathbf{y}) || q(\mathbf{z}_x))] &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} \left[ \int p(\mathbf{z}_x|\mathbf{y}) \log \frac{p(\mathbf{z}_x|\mathbf{y})p(\mathbf{z}_x|\mathbf{x})}{q(\mathbf{z}_x)p(\mathbf{z}_x|\mathbf{x})} d\mathbf{z}_x \right] \\ &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}), \mathbf{x} \sim p(\mathbf{x})} [D_{KL}(p(\mathbf{z}_x|\mathbf{y}) || p(\mathbf{z}_x|\mathbf{x}))] \\ &\quad + \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} \left[ \mathbb{E}_{\mathbf{z}_x \sim p(\mathbf{z}_x|\mathbf{y})} [\log \frac{p(\mathbf{z}_x|\mathbf{y})}{q(\mathbf{z}_x)}] \right], \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{L}(\theta_y, \varphi_y, \theta_x, \varphi_x, \theta_d, \varphi_d; \mathbf{y}) &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}), \mathbf{x} \sim p(\mathbf{x})} [D_{KL}(p(\mathbf{z}_x|\mathbf{y}) || p(\mathbf{z}_x|\mathbf{x})) \\ &\quad + \mathbb{E}_{\mathbf{z}_x \sim p(\mathbf{z}_x|\mathbf{y})} [\log \frac{p(\mathbf{z}_x|\mathbf{y})}{q(\mathbf{z}_x)}]] \\ &\quad + \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}), \mathbf{d} \sim p(\mathbf{d})} [D_{KL}(p(\mathbf{z}_d|\mathbf{y}) || p(\mathbf{z}_d|\mathbf{d})) \\ &\quad + \mathbb{E}_{\mathbf{z}_d \sim p(\mathbf{z}_d|\mathbf{y})} [\log \frac{p(\mathbf{z}_d|\mathbf{y})}{q(\mathbf{z}_d)}]] \\ &\quad - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} [\mathbb{E}_{\mathbf{z}_d, \mathbf{z}_x \sim p(\mathbf{z}_d, \mathbf{z}_x|\mathbf{y})} [\log q(\mathbf{y}|\mathbf{z}_x, \mathbf{z}_d)]]. \end{aligned} \quad (6)$$

In (5), we introduce posterior  $p(\mathbf{z}_x|\mathbf{x})$  estimated from C-VAE to conduct supervised latent variable learning. The purpose is to obtain speech latent variables from observed signal. Finally, substituting (5) into (4), the final loss function can be written as (6). In (6), we can find KL divergence constraints for speech and noise latent variables, which ensures that we can estimate the desired posterior distributions from noisy signal in a supervision way. This is also why our method can disentangle latent variables, and the noise can be estimated by nonlinear VAE rather than linear NMF, which is



**Fig. 2:** Model of Bayesian permutation training for SE.

different from the previous VAE-based SE methods [13–17]. Moreover, in (6),  $-\mathcal{L}$  can be used as a novel variational lower bound to perform supervised VAE training in other VAE-related applications. To better minimize (6), we introduce C-VAE and N-VAE to conduct joint training, which forms a Bayesian permutation training process between the three VAEs. Finally, the NS-VAE’s training loss is

$$\begin{aligned} \mathcal{L}_{total} = & \mathcal{L}(\theta_y, \varphi_y, \theta_x, \varphi_x, \theta_d, \varphi_d; \mathbf{y}) \\ & + \mathcal{L}_c(\theta_x, \varphi_x; \mathbf{x}) + \mathcal{L}_n(\theta_d, \varphi_d; \mathbf{d}), \end{aligned} \quad (7)$$

where  $\mathcal{L}_c(\theta_x, \varphi_x; \mathbf{x})$  and  $\mathcal{L}_n(\theta_d, \varphi_d; \mathbf{d})$  are the general VAE loss function for speech and noise, which can be written as

$$\mathcal{L}_c(\theta_x, \varphi_x; \mathbf{x}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \{ D_{KL} (p(\mathbf{z}_x | \mathbf{x}) || q(\mathbf{z}_x)) \} - \mathbb{E}_{\mathbf{z}_x \sim p(\mathbf{z}_x | \mathbf{x})} [\log q(\mathbf{x} | \mathbf{z}_x)], \quad (8)$$

$$\begin{aligned} \mathcal{L}_n(\theta_d, \varphi_d; \mathbf{d}) = & \mathbb{E}_{\mathbf{d} \sim p(\mathbf{d})} \{ D_{KL} (p(\mathbf{z}_d | \mathbf{d}) || q(\mathbf{z}_d)) \} \\ & - \mathbb{E}_{\mathbf{z}_d \sim p(\mathbf{z}_d | \mathbf{d})} [\log q(\mathbf{d} | \mathbf{z}_d)]. \end{aligned} \quad (9)$$

In (7), it can be found that the NS-VAE’s training also includes the training of C-VAE and N-VAE, which improves NS-VAE’s ability to disentangle latent variables. Minimizing  $\mathcal{L}_{total}$  is our final target.

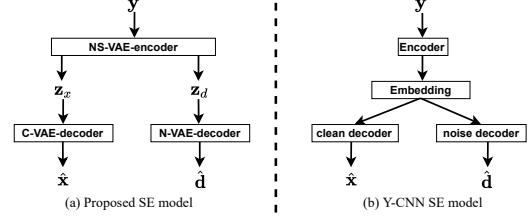
Fig. 2 shows the proposed framework. To summarize, the proposed method includes a training and an enhancement stage. The whole training process can be described as follows: first, C-VAE and N-VAE are separately pre-trained without supervision using (8) and (9). Then, the LPS features of speech, noise and observed signal are separately used as the encoder input of C-VAE, N-VAE, and NS-VAE to estimate posterior distributions  $p(\mathbf{z}_x | \mathbf{y})$ ,  $p(\mathbf{z}_d | \mathbf{y})$ ,  $p(\mathbf{z}_x | \mathbf{x})$ ,  $p(\mathbf{z}_d | \mathbf{d})$  and prior distribution  $q(\mathbf{y} | \mathbf{z}_x, \mathbf{z}_d)$ ,  $q(\mathbf{x} | \mathbf{z}_x)$ ,  $q(\mathbf{d} | \mathbf{z}_d)$ . Finally, (7) is used as a loss function to perform related parameters update with the Adam algorithm [21]. The training is completed when the neural networks converge. In the online SE stage, we assume that the  $\mathbf{z}_x$  sampled from  $p(\mathbf{z}_x | \mathbf{x})$  is approximately equal to the sample  $\mathbf{z}_x$  sampled from  $p(\mathbf{z}_x | \mathbf{y})$ . Therefore, we can separately use the NS-VAE encoder’s two outputs as input of C-VAE and N-VAE to obtain the prior distributions  $q(\mathbf{x} | \mathbf{z}_x)$  and  $q(\mathbf{d} | \mathbf{z}_d)$ . After that, using the reparameterization trick and Monte Carlo estimate (MCE) [12], the speech and noise signal can be obtained. The enhanced speech is acquired by direct waveform reconstruction or post-filtering methods. This enhanced process is shown in Fig. 3 (a).

### 3.3. Calculation of loss function

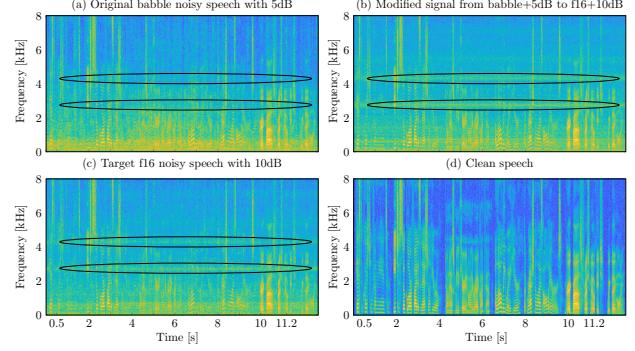
In (7), the related posterior and prior distributions need to be determined, and  $q(\mathbf{z}_x)$  and  $q(\mathbf{z}_d)$  need to be predefined for the calculation. Here, for the simplicity of calculation, we assume that all the posterior and prior distributions are multivariate normal distributions with diagonal covariance [12], which is similar to the previous VAE-based SE methods [13–17]. For the NS-VAE, we have

$$\begin{aligned} p(\mathbf{z}_x | \mathbf{y}) &= \mathcal{N} (\mathbf{z}_x; \mu_{\theta_{yx}} (\mathbf{y}), \sigma_{\theta_{yx}}^2 (\mathbf{y}) \mathbf{I}), \\ p(\mathbf{z}_d | \mathbf{y}) &= \mathcal{N} (\mathbf{z}_d; \mu_{\theta_{yd}} (\mathbf{y}), \sigma_{\theta_{yd}}^2 (\mathbf{y}) \mathbf{I}), \\ q(\mathbf{y} | \mathbf{z}_x, \mathbf{z}_d) &= \mathcal{N} (\mathbf{y}; \mu_{\varphi_y} (\mathbf{z}_x, \mathbf{z}_d), \sigma_{\varphi_y}^2 (\mathbf{z}_x, \mathbf{z}_d) \mathbf{I}), \end{aligned} \quad (10)$$

where  $\mathbf{I}$  is the identity matrix.  $\mu_{\theta_{yx}} (\mathbf{y})$ ,  $\sigma_{\theta_{yx}}^2 (\mathbf{y})$ ,  $\mu_{\theta_{yd}} (\mathbf{y})$ ,  $\sigma_{\theta_{yd}}^2 (\mathbf{y})$  can be estimated by NS-VAE’s encoder  $G_{\theta_y} (\mathbf{y})$  with parameter



**Fig. 3:** Enhancement framework comparison.



**Fig. 4:** Experiment for disentangling latent variables.

$\theta_y = \{\theta_{yx}, \theta_{yd}\}$ , and  $\sigma_{\varphi_y}^2 (\mathbf{z}_x, \mathbf{z}_d)$  and  $\mu_{\varphi_y} (\mathbf{z}_x, \mathbf{z}_d)$  can be estimated by NS-VAE’s decoder  $G_{\varphi_y} (\mathbf{z}_x, \mathbf{z}_d)$  with parameter  $\varphi_y$ . Due to the space limitation and the fact that the frameworks of C-VAE and N-VAE are similar, we only give the details of C-VAE. Here, we have

$$\begin{aligned} p(\mathbf{z}_x | \mathbf{x}) &= \mathcal{N} (\mathbf{z}_x; \mu_{\theta_x} (\mathbf{x}), \sigma_{\theta_x}^2 (\mathbf{x}) \mathbf{I}) \\ q(\mathbf{x} | \mathbf{z}_x) &= \mathcal{N} (\mathbf{x}; \mu_{\varphi_x} (\mathbf{z}_x), \sigma_{\varphi_x}^2 (\mathbf{z}_x) \mathbf{I}), \end{aligned} \quad (11)$$

where  $\mu_{\theta_x} (\mathbf{x})$ ,  $\sigma_{\theta_x}^2 (\mathbf{x})$  are obtained by C-VAE’s encoder  $G_{\theta_x} (\mathbf{x})$  with parameter  $\theta_x$ , and  $\mu_{\varphi_x} (\mathbf{z}_x)$ ,  $\sigma_{\varphi_x}^2 (\mathbf{z}_x)$  can be estimated by C-VAE’s decoder  $G_{\varphi_x} (\mathbf{z}_x)$  with parameter  $\varphi_x$ .  $q(\mathbf{z}_d)$  and  $q(\mathbf{z}_x)$  are pre-defined as a centered isotropic multivariate Gaussian  $q(\mathbf{z}_x) = \mathcal{N} (\mathbf{z}_x; \mathbf{0}, \mathbf{I})$  and  $q(\mathbf{z}_d) = \mathcal{N} (\mathbf{z}_d; \mathbf{0}, \mathbf{I})$ . Finally, when all the distributions are determined, we can apply loss function (7) and the Adam algorithm to perform related parameters update for SE.

## 4. EXPERIMENT AND RESULT ANALYSIS

In this section, the proposed algorithm is evaluated. At first, we will use an example to verify that the proposed method can disentangle different latent variables from the observed signal. After that, an experiment will show the SE performance of our method.

**Dataset:** In this work, we use the TIMIT database [22], 100 environmental noises [23], DEMAND database [24] and NOISEX-92 database [25] to evaluate the performance of the proposed algorithm. In the training stage, the Babble, F16 noise from the NOISEX-92 database [25] and 90 environmental noise (N1–N90) [23] are used to conduct experiments. All 4620 utterances from the TIMIT database are corrupted by 92 types of noise at four different signal-to-noise ratio (SNR) levels, i.e., -5, 0, 5, and 10 dB. The utterances are randomly selected from these corrupted utterances, and they are connected to a 12-hour noisy speech database. Meanwhile, the corresponding speech and noise databases are also obtained. In the test stage, 200 utterances from the TIMIT test set, including 1680 utterances, are randomly chosen to build the test database. 13 types of noise (office [24], factory [25], and 10 unseen environmental noise (N91–N100) [23]) are randomly added to the 200 utterances at four SNR levels (i.e., -5, 0, 5, and 10 dB) to conduct experiment. In our experiments, all the signals are down-sampled to 16 kHz. The frame length is 512 samples with a frame shift of 256 samples.

**Table 1:** Average PESQ and SI-SDR comparison of different methods

SNR	SI-SDR (dB)					PESQ				
	Noisy	Y-L	PVAE-L	Y-M	PVAE-M	Noisy	Y-L	PVAE-L	Y-M	PVAE-M
-5	-5.67(± 0.22)	1.25(± 0.67)	2.84(± 0.72)	2.04(± 0.68)	<b>4.01(± 0.88)</b>	1.43(± 0.02)	1.59(± 0.03)	<b>1.87(± 0.03)</b>	1.68(± 0.03)	1.86(± 0.03)
0	-0.69(± 0.22)	4.52(± 0.47)	6.32(± 0.48)	7.40(± 0.68)	<b>8.59(± 0.75)</b>	1.78(± 0.02)	2.02(± 0.02)	2.24(± 0.03)	2.11(± 0.03)	<b>2.27(± 0.03)</b>
5	4.30(± 0.23)	6.76(± 0.29)	8.67(± 0.31)	11.74(± 0.62)	<b>12.33(± 0.61)</b>	2.13(± 0.02)	2.43(± 0.02)	2.57(± 0.02)	2.53(± 0.03)	<b>2.63(± 0.02)</b>
10	7.30(± 0.23)	8.05(± 0.18)	10.03(± 0.23)	15.17(± 0.54)	<b>15.41(± 0.50)</b>	2.46(± 0.01)	2.76(± 0.02)	2.80(± 0.02)	2.86(± 0.02)	<b>2.91(± 0.02)</b>

**Baseline:** To evaluate the performance of the proposed method, we use a supervised SE model as a reference method (referred to Y-CNN) [26]. This is similar to the proposed method and can perform SE by direct waveform reconstruction [7] or estimated mask [8]. For a fair comparison, we use a convolutional neural network (CNN) to replace the original DNN [26] to improve its performance. Fig. 3 shows the framework comparison of enhancement. Y-CNN has the same encoder and decoder as the proposed model. The only difference between Y-CNN and our method is the training loss function. The loss function of the proposed method applies deep representation learning (DRL) and reasonable assumptions to disentangle latent variables, which is not achieved by Y-CNN [26].

**Experimental setups:** There are three VAEs in our proposed method. The C-VAE and N-VAE have the same structure. 1D CNN which is widely used in SE [9] is adopted in the experiment. C-VAE’s encoder includes four hidden 1D convolutional layers. The number of channels in each layer is 32, 64, 128, and 256. The size of each convolving kernel is 3. The two output layers of the encoders are fully connected layers with 128 nodes. By using the reparameterization trick, the decoders’ input size can also be set as 128. The decoder consists of four hidden 1D convolutional layers (the channel number of each layer is 256, 128, 64, and 32 with 3 kernel) and two fully connected output layers with 257 nodes. Moreover, the activation functions for the hidden and output layer are ReLU and linear activation function, respectively. For NS-VAE, its encoder also includes four 1D convolutional layers with ReLU as the activation function. The other parameter setting is the same as C-VAE. Additionally, its encoder has four output layers with 128 nodes and a linear activation function. The input size of the NS-VAE decoder is 256, which includes the latent speech and the noise variables. The decoder structure of NS-VAE’s decoder is the same as that of C-VAE (Y-CNN’s encoder and decoders only have one output layer with 128 and 257 nodes, respectively, because it does not disentangle latent variables. The other settings are the same as that of NS-VAE). In the training stage, all networks are trained by the Adam algorithm with a 128 mini-batch size. The learning rate is 0.001.

**Experimental results:** Firstly, we will verify the ability of the proposed method to effectively disentangle the speech and noise latent variables from observed signals. Based on our assumption, the observed signal  $\mathbf{y}$  is determined by  $\mathbf{z}_x$  and  $\mathbf{z}_d$ . Thus, if we use different  $\mathbf{z}_x$  or  $\mathbf{z}_d$  as the input of NS-VAE’s decoder, we can obtain the different observed signal.  $\mathbf{z}_x$  and  $\mathbf{z}_d$  can be acquired by different NS-VAE encoders. Fig. 4 shows the experimental result. In this example, we first disentangle the latent variables of the observed signal (babble noise with 5dB). Then, we keep the speech latent variable  $\mathbf{z}_x$  and replace the noise latent variable with another noise latent variable (f16 noise with 10dB). Finally, the new combination of latent variables is used as the input of NS-VAE’s decoder to acquire the modified signal. Fig. 4(b) shows the modified signal. Comparing Fig. 4(a), (b), (c), and (d), it can be found that the modified signal successfully removes the babble noise character, and the original noise character is replaced by f16 noise character. (The modified signal has the constant noise around 3000 and 4000 Hz as shown in the black circle area, which is the same as the target signal.) Further-

**Table 2:** Average STOI comparison of different methods

SNR	Noisy	Y-L	PVAE-L	Y-M	PVAE-M
-5	57.62(± 1.31)	57.63(± 1.67)	60.00(± 1.33)	59.72(± 1.70)	60.32(± 1.40)
0	70.02(± 1.24)	69.80(± 1.48)	70.68(± 1.12)	72.02(± 1.43)	71.75(± 1.19)
5	80.20(± 0.90)	79.20(± 1.18)	79.87(± 0.87)	81.96(± 1.02)	80.78(± 0.91)
10	86.32(± 0.50)	85.60(± 0.72)	84.32(± 0.54)	88.80(± 0.63)	87.24(± 0.58)

more, the modified signal also preserves the original speech character. Therefore, this experiment indicates that the proposed method can effectively disentangle different latent variables.

In the second experiment, all algorithms are evaluated by the scale-invariant signal-to-distortion ratio (SI-SDR) in decibel (dB) [27], short-time objective intelligibility (STOI) [28], and perceptual evaluation of speech quality (PESQ) [29]. The enhanced speech is obtained by direct waveform reconstruction [7] or soft time-frequency mask estimation [26]. We use Y-M and Y-L to represent that the enhanced speech is acquired by mask estimation and direct waveform reconstruction using Y-CNN, respectively. Similarly, PVAE-L and PVAE-M denote that the enhanced speech is obtained by the proposed method using direct waveform reconstruction and mask estimation, respectively. Table. 1 and 2 show the PESQ, SI-SDR, and STOI comparisons with a 95% confidence interval. The results verify that our method can learn latent speech and noise variables from observed signals because PVAE-L achieves better PESQ and SI-SDR performance than Y-L. This means that C-VAE’s decoder can recognize speech latent variables that are disentangled by NS-VAE. Moreover, PVAE-M significantly achieves better PESQ and SI-SDR performance than Y-M, which shows that our method can estimate a more accurate mask for SE. This result also illustrates that our approach has better noise estimation performance than the reference method. Additionally, the results also show that Y-CNN’s performance can be improved by the proposed loss function. Table. 2 shows that the STOI score is competitive between the proposed and the reference algorithms. We think that the STOI score of the proposed method can be further improved by improving PVAE’s disentangling performance [30]. Overall, PVAE-M achieves the best SE performance across the three evaluation criteria. Here, we only use a basic neural network to verify our algorithm. Its performance can be further improved by using more advanced neural networks and other speech features [2, 9, 10].

## 5. CONCLUSION

In this paper, a supervised Bayesian permutation training DRL method is proposed to disentangle latent speech and noise variables from the observed signal for SE. The proposed method is based on VAE and Bayesian theory. The experimental results show that our method cannot only successfully disentangle different latent variables but also obtain higher SI-SDR and PESQ scores than the state-of-the-art reference method. Moreover, the results also illustrate that the SE performance of the reference method can be improved by introducing the proposed DRL algorithm. In future work, some other strategies can be considered to further improve the disentangling performance of latent variables. In addition, the proposed method can also be applied in other speech generative tasks, e.g., voice conversion and ASR.

## 6. REFERENCES

- [1] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.
- [2] D. Wang and J. Chen, “Supervised speech separation based on deep learning: An overview,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, 2018.
- [3] T. Gerkmann and R. C. Hendriks, “Unbiased MMSE-based noise power estimation with low complexity and low tracking delay,” *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 20, no. 4, pp. 1383–1393, 2011.
- [4] J. R. Jensen, J. Benesty, and M. G. Christensen, “Noise reduction with optimal variable span linear filters,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 24, no. 4, pp. 631–644, 2015.
- [5] Y. Xiang, L. Shi, J. L. Højvang, M. H. Rasmussen, and M. G. Christensen, “An NMF-HMM speech enhancement method based on kullback-leibler divergence,” in *Proc. Interspeech*, 2020, pp. 2667–2671.
- [6] M. S. Kavalekalam, J. K. Nielsen, L. Shi, M. G. Christensen, and J. Boldt, “Online parametric NMF for speech enhancement,” in *Proc. European Signal Processing Conf.*, 2018, pp. 2320–2324.
- [7] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 23, no. 1, pp. 7–19, 2014.
- [8] Y. Wang, A. Narayanan, and D. Wang, “On training targets for supervised speech separation,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [9] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [10] A. Li, W. Liu, X. Luo, C. Zheng, and X. Li, “Icassp 2021 deep noise suppression challenge: Decoupling magnitude and phase optimization with a two-stage deep network,” *arXiv preprint arXiv:2102.04198*, 2021.
- [11] Y. Xiang and C. Bao, “A parallel-data-free speech enhancement method using multi-objective learning cycle-consistent generative adversarial network,” *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 28, pp. 1826–1838, 2020.
- [12] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [13] 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.*, 2018, pp. 716–720.
- [14] S. Leglaive, L. Girin, and R. Horraud, “A variance modeling framework based on variational autoencoders for speech enhancement,” in *Proc. IEEE Workshop Machine Learning. Signal Process.*, 2018, pp. 1–6.
- [15] ———, “Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 101–105.
- [16] G. Carbajal, J. Richter, and T. Gerkmann, “Guided variational autoencoder for speech enhancement with a supervised classifier,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 681–685.
- [17] H. Fang, G. Carbajal, S. Wermter, and T. Gerkmann, “Variational autoencoder for speech enhancement with a noise-aware encoder,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 676–680.
- [18] W.-N. Hsu, Y. Zhang, and J. Glass, “Learning latent representations for speech generation and transformation,” *Proc. Interspeech*, pp. 1273–1277, 2017.
- [19] ———, “Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation,” in *Proc. IEEE Workshop Automatic. Speech Recognition. and Understanding.*, 2017, pp. 16–23.
- [20] ———, “Unsupervised learning of disentangled and interpretable representations from sequential data,” in *Proc. Advances in Neural Inform. Process. Syst.*, 2017, pp. 1876–1887.
- [21] D. P. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [22] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1,” *NASA STI/Recon technical report n*, vol. 93, 1993.
- [23] G. Hu and D. Wang, “A tandem algorithm for pitch estimation and voiced speech segregation,” *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 18, no. 8, pp. 2067–2079, 2010.
- [24] J. Thiemann, N. Ito, and E. Vincent, “The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings,” in *Proceedings of Meetings on Acoustics ICA2013*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.
- [25] A. Varga and H. J. Steeneken, “Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems,” *Speech Commun.*, vol. 12, no. 3, pp. 247–251, 1993.
- [26] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Deep learning for monaural speech separation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1562–1566.
- [27] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “Sdr—half-baked or well done?” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 626–630.
- [28] 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, and Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.
- [29] 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 Process.*, vol. 2, 2001, pp. 749–752.
- [30] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, “Understanding disentangling in  $\beta$ -vae,” *arXiv preprint arXiv:1804.03599*, 2018.