# A Statistically Principled and Computationally Efficient Approach to Speech Enhancement using Variational Autoencoders

Manuel Pariente, Antoine Deleforge, Emmanuel Vincent

# A Statistically Principled and Computationally Efficient Approach to Speech Enhancement using Variational Autoencoders

*Manuel Pariente*      *Antoine Deleforge*      *Emmanuel Vincent*

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

manuel.pariente@univ-lorraine.fr, {antoine.deleforge, emmanuel.vincent}@inria.fr

## Abstract

Recent studies have explored the use of deep generative models of speech spectra based on variational autoencoders (VAEs), combined with unsupervised noise models, to perform speech enhancement. These studies developed iterative algorithms involving either Gibbs sampling or gradient descent at each step, making them computationally expensive. This paper proposes a variational inference method to iteratively estimate the power spectrogram of the clean speech. Our main contribution is the analytical derivation of the variational steps in which the encoder of the pre-learned VAE can be used to estimate the variational approximation of the true posterior distribution, using the very same assumption made to train VAEs. Experiments show that the proposed method produces results on par with the aforementioned iterative methods using sampling, while decreasing the computational cost by a factor 36 to reach a given performance.

**Index Terms**: Speech enhancement, variational autoencoders, variational Bayes, non-negative matrix factorization.

## 1. Introduction

Speech enhancement is the problem of extracting a speech source from a noisy recording [1, 2]. It is commonly tackled by discriminative methods using deep neural networks (DNNs) to predict time-frequency masks [3, 4], clean power spectrograms [5] or to model the signal variance [6] from spectrograms of mixtures. While the performances of these methods are satisfactory, their generalization capabilities are limited due to their supervised nature. Indeed, DNNs trained this way are specific to a task and a recording condition. Instead, generative approaches do not suffer from this drawback. Once a source model has been learned, the statistical model can account for different tasks and recording conditions.

Non-negative matrix factorization (NMF) [7] is a popular method for speech enhancement in which power or amplitude spectrograms are approximated as the product of two non-negative matrices. While certain forms of NMF can be interpreted as generative, they suffer from their linearity assumptions. Very recent studies have combined the modelling power of DNN with statistical approaches in order to design unsupervised speech enhancement and source separation methods in both single-channel [8–10] and multichannel scenarios [11–15]. The main idea of these studies is to use variational autoencoders (VAEs) [16] to replace the NMF generative model, thus

benefiting from DNNs' representational power. Combined with an observation model, Expectation-Maximization (EM) [17] or Bayesian inference [18] algorithms can be derived to iteratively estimate individual source spectra from a given mixture.

In [8–12], speech enhancement is performed using a pre-trained VAE-based generative model of speech spectra combined with an unsupervised NMF model for the noise. Inference of the clean speech involves using the Metropolis-Hastings algorithm [19] to estimate an intractable distribution over the VAE's latent space. Similar approaches applied to source separation are developed in [13, 14], where the latent variables are updated using backpropagation. While the algorithms analytically derived in [8–15] show promising performances, sampling and backpropagation methods are computationally expensive.

In order to train a VAE, an encoder is jointly learned with the generative model. The role of the encoder is to approximate the true, but intractable, posterior. Once the VAE is trained, the encoder could in principle be used to approximate the true posterior. Interestingly, [15] proposes a heuristic algorithm for multichannel source separation based on [13] which uses this property of the encoder to achieve computational efficiency. However this inference algorithm is not statistically principled.

In this study, we present a Bayesian single-channel speech enhancement algorithm in which a VAE is used as generative model of speech spectra, and the noise is modelled using NMF. After unsupervised training of the speech model, the inference consists in iteratively updating the source estimates using Wiener filtering and updating the corresponding latent representation using the encoder. No sampling or backpropagation being required, our method is significantly faster than in [8–14]. The use of the encoder differs from [15], and, unlike it, is not heuristic but motivated by the same variational approximation as the one used to train VAEs. To the best of our knowledge, this is the first method to propose a statistically principled approach re-using the probabilistic encoder as a posterior approximator.

We first introduce the VAE framework and its application to spectrogram modelling. We then detail our assumptions and our inference algorithm. Experiments and results will be presented before we finally conclude.

## 2. Model

Given an observation of clean speech embedded in noise, the goal of speech enhancement is to produce an estimate of the clean speech. Our approach is to use a VAE as the generative model of clean speech spectra to infer this estimate. We will first describe the VAE framework and its training procedure in Section 2.1 and 2.2 before introducing the observation model in Section 2.3.

## 2.1. Generative speech model

A common approach to modeling sound sources is to assume the short time Fourier Transform (STFT) coefficients are proper complex Gaussian random variables. With $f$ denoting the frequency index and $t$ the time-frame index, for all $(f, t) \in \{0, ..., F-1\} \times \{0, ..., N-1\}$, each source STFT coefficient $s_{ft} \in \mathbb{C}$ independently follows

$$s_{ft} \sim \mathcal{N}_c(0, \sigma_{ft}^2). \tag{1}$$

Several frameworks have been used to model the variance $\sigma_{ft}^2$ of this distribution [2, 20]. To this end, we choose to use the VAE framework, which uses the representational power of DNNs to build generative models. It consists in mapping a zero-mean unit Gaussian random variable $\boldsymbol{z}_t \in \mathbb{R}^L$ through a DNN, into the desired output distribution. In our setting, we use it to represent $\sigma_{ft}^2$. The generative model can be written as

$$\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{2}$$

$$s_{ft}|\boldsymbol{z}_t; \theta \sim \mathcal{N}_c(0, \sigma_f^2(\boldsymbol{z}_t)), \tag{3}$$

where $\sigma_f^2 : \mathbb{R}^L \mapsto \mathbb{R}_+$ is a non-linear function implemented by a DNN with parameters $\boldsymbol{\theta}$. Each $\sigma_f^2$ is one neuron of a F-dimensional output layer of a single DNN.

## 2.2. VAE training and posterior approximation

The VAE is trained on clean speech data by maximizing the likelihood $p_\theta(\boldsymbol{s}) = \prod_t p_\theta(\boldsymbol{s}_t)$, with $\boldsymbol{s}_t = [s_{1t}, ..., s_{Ft}]^T$. To do so, we introduce a variational approximation of $p_\theta(\boldsymbol{z}_t|\boldsymbol{s}_t)$, $q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_t)$, parametrized by $\phi$. We have

$$\log p_\theta(\boldsymbol{s}_t) = \mathcal{D}_{KL}(q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_t)||p_\theta(\boldsymbol{z}_t|\boldsymbol{s}_t)) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}_t) \tag{4}$$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}_t) = \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{s}_t|\boldsymbol{z}_t)] - \mathcal{D}_{KL}(q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_t)||p(\boldsymbol{z}_t)), \tag{5}$$

where $\mathcal{D}_{KL}(q||p) = -\mathbb{E}_q[\log(p/q)]$ denotes the Kullback-Leibler (KL) divergence. The KL term on the left hand side of (4) being always positive, $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}_t)$ is a lower bound of the marginal log-likelihood $\log p_\theta(\boldsymbol{s}_t)$. We can thus maximize the individual $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}_t)$ with respect to $\boldsymbol{\theta}$ and $\phi$ in order to maximize the log-likelihood $\log p_\theta(\boldsymbol{s})$. For all $(l, t) \in \{0, ..., L-1\} \times \{0, ..., N-1\}$, $q_\phi(z_{lt}|\boldsymbol{s}_t)$ is defined as in [8]

$$z_{l,t}|\boldsymbol{s}_t; \phi \sim \mathcal{N}(\tilde{\mu}_l(|\boldsymbol{s}_t|^2), \tilde{\sigma}_l^2(|\boldsymbol{s}_t|^2)), \tag{6}$$

where $\tilde{\mu}_l$ and $\tilde{\sigma}_l^2$ are DNN-based functions parametrized by $\phi$. Finally, we can use (2), (3) and (6) to write (5) as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}) \overset{c}{=} \sum_f \mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_{ft})} d_{IS}(|s_{ft}|^2, \sigma_f^2(\boldsymbol{z}_t)) \tag{7}$$

$$+ \frac{1}{2} \sum_l \log(\tilde{\sigma}_l^2(|\boldsymbol{s}_t|^2)) - \tilde{\mu}_l^2(|\boldsymbol{s}_t|^2) - \tilde{\sigma}_l^2(|\boldsymbol{s}_t|^2),$$

where $d_{IS}(x, y) = x/y - \log(x/y) - 1$ denotes the Itakura-Saito divergence, $\boldsymbol{z}_t = [z_{1t}, ..., z_{Lt}]^T$ and $\overset{c}{=}$ denotes equality up to a constant. Using the so-called reparametrization trick [16], we can maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s})$ using gradient descent optimization algorithms [21].

The variational distribution $q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_t)$ is usually considered as an unwanted by-product of the training procedure of the

VAE, only introduced to learn $p_\theta(\boldsymbol{s}_t|\boldsymbol{z}_t)$. However, by examining carefully (4), we can see that while maximizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{s}_t)$ can only increase the log-likelihood, it also decreases the KL divergence between the true posterior and its variational approximation. In other words, in addition to the generative speech model, the VAE provides a variational approximation of the true posterior. This feature will play a key role in the inference algorithm proposed in Section 3.

## 2.3. Noisy speech model

As in [8], the STFT coefficients $n_{ft}$ of the noise are modelled with a rank-$K$ NMF Gaussian model. For all $(f, t)$

$$n_{ft} \sim \mathcal{N}_c(0, (\boldsymbol{WH})_{ft}), \tag{8}$$

with $\boldsymbol{W} \in \mathbb{R}_+^{F \times K}$, $\boldsymbol{H} \in \mathbb{R}_+^{K \times N}$.

Finally, independently for all $(f, t)$, the single-channel observation of the noisy speech is modeled by

$$x_{ft} = s_{ft} + n_{ft}, \tag{9}$$

where $s_{ft}$ and $n_{ft}$ are independent.

# 3. Inference

Given a noisy speech signal $\boldsymbol{X} = \{x_{ft}\}_{(ft)}$, our goal is now to maximize the likelihood of $\boldsymbol{X}$ given the mixture model (9), the generative model of speech (2), (3) and the NMF model (8). With $\boldsymbol{n}_t = [n_{1t}, ..., n_{Ft}]^T$ and $\boldsymbol{H}_t = [H_{1t}, ..., H_{Kt}]^T$, we consider $\boldsymbol{y}_t = \{\boldsymbol{s}_t, \boldsymbol{n}_t, \boldsymbol{z}_t\}$ to be the set of latent variables and $\boldsymbol{\Theta}_t = \{\boldsymbol{W}, \boldsymbol{H}_t\}$ the parameters of the model. We introduce $r(\boldsymbol{y}_t)$, a variational approximation of $p(\boldsymbol{y}_t|\boldsymbol{x}_t; \boldsymbol{\Theta}_t)$. We have

$$\log p(\boldsymbol{x}_t; \boldsymbol{\Theta}_t) = \mathcal{D}_{KL}(r(\boldsymbol{y}_t)||p(\boldsymbol{y}_t|\boldsymbol{x}_t; \boldsymbol{\Theta}_t)) + \mathcal{L}(r, \boldsymbol{\Theta}_t)$$

$$\mathcal{L}(r, \boldsymbol{\Theta}_t) = \mathbb{E}_{r(\boldsymbol{y}_t; \boldsymbol{\Theta})} \left[ \log \frac{p(\boldsymbol{x}_t, \boldsymbol{y}_t; \boldsymbol{\Theta}_t)}{r(\boldsymbol{y}_t)} \right]. \tag{10}$$

We suppose that the variational distribution $r(\boldsymbol{y}_t)$ factorizes as

$$r(\boldsymbol{s}_t, \boldsymbol{n}_t, \boldsymbol{z}_t) = r(\boldsymbol{s}_t, \boldsymbol{n}_t)r(\boldsymbol{z}_t) = \prod_f r(s_{ft}, n_{ft}) \prod_l r(z_{lt}). \tag{11}$$

We can then iteratively maximize $\mathcal{L}(r, \boldsymbol{\Theta}_t)$ with respect to the factorized distributions and the NMF parameters. The variational distributions' updates are given by (10.9) in [18]

$$\log r(s_{ft}, n_{ft}) \overset{c}{=} \mathbb{E}_{r(\boldsymbol{z}_t)} \log p(x_{ft}, s_{ft}, n_{ft}, \boldsymbol{z}_t; \boldsymbol{\Theta}_{ft}) \tag{12}$$

$$\log r(z_{lt}) \overset{c}{=} \mathbb{E}_{r(\boldsymbol{s}_t, \boldsymbol{n}_t)} \log p(\boldsymbol{x}_t, \boldsymbol{s}_t, \boldsymbol{n}_t, z_{lt}; \boldsymbol{\Theta}_t). \tag{13}$$

$\boldsymbol{W}$ and $\boldsymbol{H}$ can be updated by maximizing the following:

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{old}) \overset{c}{=} \mathbb{E}_{r(\boldsymbol{y}; \boldsymbol{\Theta}^{old})}[\log p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\Theta})]. \tag{14}$$

We detail the updates of this Variational Expectation-Maximization (VEM) algorithm below, and in the supporting document [22].

## 3.1. E-(s,n) step

We define $\sigma_{n,ft}^2 = (\boldsymbol{WH})_{ft}$ to make the notation less cluttered. Using (12), we find (see [22])

$$r(\boldsymbol{s}_t, \boldsymbol{n}_t) \sim \prod_f \mathcal{N}_c(\boldsymbol{\mu}_{ft}, \boldsymbol{\Sigma}_{ft}) = \mathcal{N}_c(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \qquad (15)$$

where $\boldsymbol{\mu}_{ft}$ and $\boldsymbol{\Sigma}_{ft}$ are defined as

$$\boldsymbol{\Sigma}_{ft} = \boldsymbol{\Sigma}_{ss,ft} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \boldsymbol{\mu}_{ft} = \frac{x_{ft}}{\gamma_{ft}^2 + \sigma_{n,ft}^2} \begin{bmatrix} \gamma_{ft}^2 \\ \sigma_{n,ft}^2 \end{bmatrix}, \qquad (16)$$

with

$$\boldsymbol{\Sigma}_{ss,ft} = \frac{\gamma_{ft}^2 \sigma_{n,ft}^2}{\gamma_{ft}^2 + \sigma_{n,ft}^2} \qquad (17)$$

and

$$\frac{1}{\gamma_{ft}^2} = \mathbb{E}_{r(\boldsymbol{z}_t)} \left[ 1/\sigma_f^2(\boldsymbol{z}_t) \right] \approx \sum_{d=1}^D \left[ 1/\sigma_f^2(\boldsymbol{z}_t^{(d)}) \right], \qquad (18)$$

where $\{\boldsymbol{z}_n^{(d)}\}_{d=1,..,D}$ are randomly drawn from $r(\boldsymbol{z}_t)$.

### 3.2. E-z step

We can compute $r(\boldsymbol{z}_t)$ using (13) (see [22]), we get

$$\log r(\boldsymbol{z}_t) \stackrel{c}{=} \log p(\boldsymbol{z}_t) + \log p(\boldsymbol{s}_t = \left(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t}\right)^{\frac{1}{2}} |\boldsymbol{z}_t)$$

$$= \log p(\boldsymbol{z}_t | \boldsymbol{s}_t = \left(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t}\right)^{\frac{1}{2}}), \qquad (19)$$

where (19) was obtained using Bayes' theorem. The VAE assumes that $p_\theta(\boldsymbol{z}_t|\boldsymbol{s}_t)$ can be appproximated by $q_\phi(\boldsymbol{z}_t|\boldsymbol{s}_t)$. We further assume that this still holds for $\boldsymbol{s}_t$ of the form $\left(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t}\right)^{\frac{1}{2}}$. That is to say

$$r(z_{lt}) \approx q_\phi(z_{lt} | \boldsymbol{s}_t = \left(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t}\right)^{\frac{1}{2}}) \qquad (20)$$

$$= \mathcal{N}(\tilde{\mu}_l(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t}), \tilde{\sigma}_l^2(|\boldsymbol{\mu}_{s,t}|^2 + \boldsymbol{\Sigma}_{ss,t})). \qquad (21)$$

### 3.3. M-step

We now maximize $\mathcal{L}(r, \boldsymbol{\Theta}_t)$ with respect to $\boldsymbol{\Theta}_t$ using (14). With $\boldsymbol{V} \in \mathbb{R}_+^{F \times N}$ defined as $(\boldsymbol{V})_t = |\boldsymbol{\mu}_{n,t}|^2 + \boldsymbol{\Sigma}_{nn,t}$, and $\odot$ denoting element-wise matrix multiplication and exponentiation, we obtain multiplicative updates as in [23]:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^T \left((\boldsymbol{W}\boldsymbol{H})^{\odot-2} \odot \boldsymbol{V}\right)}{\boldsymbol{W}^T(\boldsymbol{W}\boldsymbol{H})^{\odot-1}}, \qquad (22)$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\left((\boldsymbol{W}\boldsymbol{H})^{\odot-2} \odot \boldsymbol{V}\right)\boldsymbol{H}^T}{(\boldsymbol{W}\boldsymbol{H})^{\odot-1}\boldsymbol{H}^T}. \qquad (23)$$

### 3.4. Speech Reconstruction

Let $\boldsymbol{\Theta}^\star = \{\boldsymbol{W}^\star, \boldsymbol{H}^\star\}$ and $r^\star(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{z}) = r^\star(\boldsymbol{s}, \boldsymbol{b})r^\star(\boldsymbol{z})$ be respectively the set of NMF parameters and the variational distribution estimated by the proposed algorithm. The final estimate of the source is given, as in [8], by

$$\hat{s}_{ft} = \mathbb{E}_{p(s_{ft}|x_{ft};\boldsymbol{\Theta}^\star)}[s_{ft}] \qquad (24)$$

$$= \mathbb{E}_{p(\boldsymbol{z}_t|x_{ft};\boldsymbol{\Theta}^\star)} \left[ \frac{\sigma_f^2(\boldsymbol{z}_t)}{\sigma_f^2(\boldsymbol{z}_t) + (\boldsymbol{W}\boldsymbol{H})_{ft}} \right] x_{ft}. \qquad (25)$$

There are three ways to evaluate this estimate. The straightforward way is to replace $p(s_{ft}|x_{ft}; \boldsymbol{\Theta}^\star)$ in (24) by its variational approximation $r^\star(s_{ft})$. The expectation over $p(\boldsymbol{z}_t|x_{ft}; \boldsymbol{\Theta}^\star)$ can also be approximated using the Metropolis Hastings (MH) algorithm as in [8], using the mean of $r^\star(\boldsymbol{z}_t)$ as the initial sample. And finally we can compute the expectation in (25) by replacing $p(\boldsymbol{z}_t|x_{ft}; \boldsymbol{\Theta}^\star)$ by its variational approximation $r^\star(\boldsymbol{z}_t)$. We will respectively refer to these methods as *S-Wiener*, *MH-Wiener* and *Z-Wiener* for ease of referencing.

*Metropolis-Hastings :* At the $m$-th iteration of the Metropolis-Hastings algorithm, we draw a random sample for each $t$ according to

$$\boldsymbol{z}_t|\boldsymbol{z}_t^{(m-1)}; \epsilon_{mh}^2 \sim \mathcal{N}(\boldsymbol{z}_t^{(m-1)}, \epsilon_{mh}^2\mathbf{I}). \qquad (26)$$

The acceptance probability can be computed as

$$\alpha = \min\left(1, \frac{p(\boldsymbol{x}_t|\boldsymbol{z}_t; \boldsymbol{\Theta}^\star)p(\boldsymbol{z}_t)}{p(\boldsymbol{x}_t|\boldsymbol{z}_t^{(m-1)}; \boldsymbol{\Theta}^\star)p(\boldsymbol{z}_t^{(m-1)})}\right). \qquad (27)$$

We accept the new sample and set $\boldsymbol{z}_t^{(m)} = \boldsymbol{z}_t$ only if $u$, drawn from a uniform distribution $\mathcal{U}([0,1])$, is smaller than $\alpha$. We keep only the last $R$ samples to compute $\hat{s}_{ft}$ from (25) and discard the samples drawn during the burn-in period.

## 4. Experimental evaluation

### 4.1. Experimental settings

*Dataset*: As in [8, 9], we use the the TIMIT corpus [24] to train the clean speech model. At inference time, we mix speech signals from the TIMIT test set with noise signals from the DE-MAND corpus [25] at 0 dB signal-to-noise ratio, according to the file provided in in the original implementation of [8] [1]. Note that both speakers and utterances are different from those in the training set.

*VAE's architecture and training*: The architecture of the VAE is the same as in [8]. The hyperbolic tangent of an input 128-dimensional linear layer passes through two L-dimensional linear layers to estimate $\tilde{\mu}_l(|\boldsymbol{s}_t|^2)$ and $\log \tilde{\sigma}_l^2(|\boldsymbol{s}_t|^2)$, from which $\boldsymbol{z}_t$ is sampled using the reparametrization trick. $\boldsymbol{z}_t$ is then fed to a 128-dimensional linear layer with hyperbolic tangent activation followed by an linear output layer predicting $\log \sigma_f^2(\boldsymbol{z}_t)$. For training, we computed the STFT of the utterances with a 1024-point sine window and a hop size of 256 points. 20% of the training set is held for validation. A VAE with $L = 64$ is trained with the Adam optimizer [26], using a learning rate of $10^{-3}$ and a batch size of 128. Training stops if no improvement is seen on the validation loss for 10 epochs.

*Baseline methods*: We compare our method to two baselines. The first baseline, developed in [8], shares the same statistical assumptions made in Section 2 except for the inclusion of a gain factor in (9). The inference is performed using a Monte Carlo Expectation-Maximization (MCEM) algorithm in which MH sampling is used to estimate the true posterior. We will refer to this method as *MCEM*. The second baseline is a heuristic we introduce to evaluate the impact of the covariance term $\boldsymbol{\Sigma}_{ss,t}$ in (21) on the convergence of the algorithm. We simply remove $\boldsymbol{\Sigma}_{ss,t}$ from (21) and the rest of the algorithm remains unchanged. Note that this is similar to the heuristic developped in [15] for determined multichannel source separation, only adapted to single-channel speech enhancement.
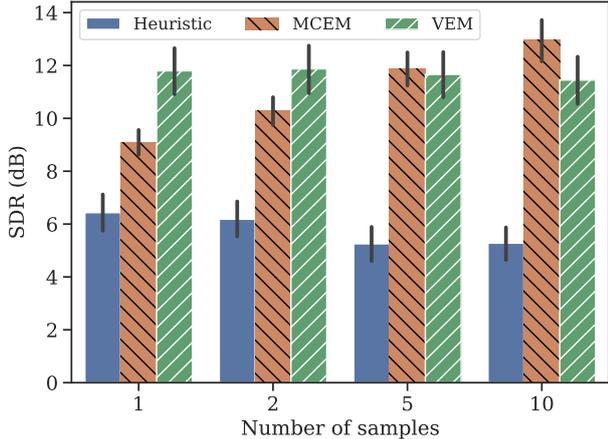
---

[1]https://gitlab.inria.fr/sileglai/mlsp2018

Figure 1: *SDR as a function of number of samples M for an input SNR of 0 dB. using MH for reconstruction. Heuristic, MCEM and VEM methods are defined in Section 4.1. Errors bars indicate 95% confidence interval.*

*Inference Setting*: For fair comparaison, we use the same settings for the three methods : the rank of the NMF model is set to $K = 10$, the latent dimension of the VAE is set to $L = 64$ and the same stopping criterion is used as in [8]. $\boldsymbol{z}_t$ is initialized based on $q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and for the final speech estimate, equation (25) is approximated using the last 25 samples of a 100-iteration MH sampling with $\epsilon^2 = 0.01$. In [8], for $R$ samples used, the total number of draws is $4R$. The methods will be compared for a same number of samples actually used to estimate the expected values : if $D$ samples are used to evaluate (18), $4D$ samples will be drawn in the MCEM case.

### 4.2. Results

*Experiment 1*: We first compare the performances of the three methods on the test set described above, in terms of Signal to Distorsion Ration (SDR) [27], computed using the *mir_eval*[2] toolbox. The comparison is done for different numbers of samples $D \in \{1, 2, 5, 10\}$. As can be seen in Fig. 1, the heuristic algorithm consistently performs worse than both other methods, proving the importance of the covariance term $\boldsymbol{\Sigma}_{ss,t}$ in (21). We also see that the performance of the proposed algorithm does not depend on the number of samples drawn to approximate (18), contrary to the MCEM approach in which performance improves as the number of samples increases. Both the factorization in (11) for VEM and the Monte Carlo approximation for MCEM introduce approximation errors, but as the number of samples increases the approximation error caused by Monte Carlo decreases. This explains why VEM achieves best performance for $R = 1$, and MCEM for $R = 10$.

*Experiment 2*: We then compare the SDR achieved by the MCEM and VEM methods as a function of the computation time, in terms of number of iterations and absolute time. The number of samples $D$ was set to 1 for the VEM, and $R$ to 5 for the MCEM so as to achieve a similar SDR at convergence (see Fig. 1). At each iteration, we estimate the speech spectra by approximating (25) with MH sampling and we compute the SDR. This is done for each test utterance. Each individual SDR curve is then padded with the SDR obtained at the last iteration
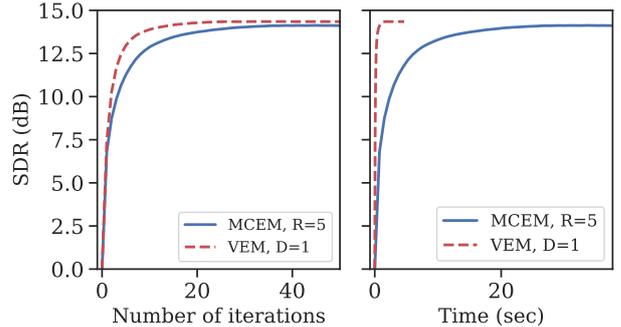
[2]https://github.com/craffel/mir_eval



Figure 2: *Average convergence speed. SDR as a function of (left) the number of iterations and (right) computation time.*

and all SDR curves are averaged to produce the left graph in Fig. 2. We observe that the proposed method converges faster which suggests that using the encoder allows for bigger jumps in the latent space than the sampling method with a small number of samples. On a 4-core i7-8650U CPU, one iteration takes on average 55 ms and 753 ms for the VEM and the MCEM algorithms, respectively. Using these numbers, we can convert the SDR as a function of the number of iterations to the SDR as a function of time. This is shown in the right part of Fig. 2 where the superiority of the proposed algorithmm in terms of absolute speed is clear. To compare the execution times, for each utterance and for both algorithms we compute the number of iterations needed to reach the final SDR up to 0.5 dB tolerance. The ratio between the number of iterations multiplied by the time per iteration ratio yields an average computational cost decrease factor of 36 to reach the same performance.

*Experiment 3*: In this experiment, we compare the three different ways of computing the source STFT coefficients in (24). We present the mean SDR after convergence for the three possible methods in Table 1. We can see that MH sampling consistently outperforms the other methods, suggesting that a better posterior approximation benefits to the quality of the reconstruction.

| | MH-Wiener | S-Wiener | Z-Wiener |
|---|---|---|---|
| VEM | **11.8** | 11.4 | 11.4 |
| Heuristic | **6.4** | 5.8 | 5.8 |
| MCEM | 11.9 | / | / |

Table 1: *SDR (dB) as a function of the reconstruction method. MH-Wiener, S-Wiener and Z-Wiener methods are defined in Section 3.4*

## 5. Conclusion

In this paper, we proposed a speech enhancement method based on a variational EM algorithm. Our main contribution is the analytical derivation of the variational steps in which the encoder of the pre-learned VAE can be used to estimate the variational approximation of the true posterior distribution, using the very same assumption made to train VAEs. Experimental results showed that the principled approach outperforms its heuristic counterpart, and produces results on par with the algorithm proposed by Leglaive et al. [8] in which the true posterior is approximated using MH sampling. Additionally, the proposed algorithm converges 36 times faster. Future work includes multichannel and multisource extension of this work, and the use of other statistical models of speech spectra [28–30].

# 6. References

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. Boca Raton, FL, USA: CRC Press, Inc., 2013.

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

[3] F. Weninger, J. R. Hershey, J. Le Roux, and B. W. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 577–581.

[4] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: discriminative embeddings for segmentation and separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2016, pp. 31–35.

[5] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *INTERSPEECH*, 2013, pp. 436–440.

[6] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[7] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[8] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.

[9] S. Leglaive, U. Simsekli, A. Liutkus, L. Girin, and R. Horaud, "Speech enhancement with variational autoencoders and alpha-stable distributions," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2019, pp. 1–5.

[10] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 716–720.

[11] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 101–105.

[12] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian Multichannel Speech Enhancement with a Deep Speech Prior," in *APSIPA*, 2018, pp. 1233–1239.

[13] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Semi-blind source separation with multichannel variational autoencoder," *arXiv preprint arXiv:1808.00892*, Aug 2018.

[14] S. Seki, H. Kameoka, L. Li, T. Toda, and K. Takeda, "Generalized multichannel variational autoencoder for underdetermined source separation," *arXiv preprint arXiv:1810.00223*, Sep 2018.

[15] L. Li, H. Kameoka, and S. Makino, "Fast MVAE joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," *CoRR*, vol. abs/1812.0, 2018.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *ICLR*, 2014.

[17] G. Wei and M. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1 1990.

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

[19] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2005.

[20] E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, "Probabilistic modeling paradigms for audio source separation," in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. IGI Global, 2010, pp. 162–185.

[21] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1600.04747*, 2016.

[22] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders : Supporting document," Inria, Tech. Rep. RR-9268, 2019. [Online]. Available: https://hal.inria.fr/hal-02089062

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

[24] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic phonetic continuous speech corpus," *Linguistic data consortium*, 1993.

[25] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings," in *Proc. Int. Cong. on Acoust.*, 2013, p. 3591.

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

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

[28] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A deep generative model of speech complex spectrograms," *arXiv preprint arXiv:1903.03269*, Nov 2019.

[29] P. Magron and T. Virtanen, "Bayesian anisotropic gaussian model for audio source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 166–170.

[30] A. Liutkus, C. Rohlfing, and A. Deleforge, "Audio source separation with magnitude priors: The BEADS model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 56–60.