Unseen Noise Estimation Using Separable Deep Auto Encoder for Speech Enhancement

Meng Sun*, *Member, IEEE,* Xiongwei Zhang, Hugo Van hamme, *Senior Member, IEEE,*
Thomas Fang Zheng, *Senior Member, IEEE*

***Abstract*—Unseen noise estimation is a key yet challenging step to make a speech enhancement algorithm work in adverse environments. At worst, the only prior knowledge we know about the encountered noise is that it is *different* from the involved speech. Therefore, by subtracting the components which cannot be adequately represented by a well defined speech model, the noises can be estimated and removed. Given the good performance of deep learning in signal representation, a deep auto encoder (DAE) is employed in this work for accurately modeling the clean speech spectrum. In the subsequent stage of speech enhancement, an extra DAE is introduced to represent the residual part obtained by subtracting the estimated clean speech spectrum (by using the pre-trained DAE) from the noisy speech spectrum. By adjusting the estimated clean speech spectrum and the unknown parameters of the noise DAE, one can reach a stationary point to minimize the total reconstruction error of the noisy speech spectrum. The enhanced speech signal is thus obtained by transforming the estimated clean speech spectrum back into time domain. The above proposed technique is called separable deep auto encoder (SDAE). Given the under-determined nature of the above optimization problem, the clean speech reconstruction is confined in the convex hull spanned by a pre-trained speech dictionary. New learning algorithms are investigated to respect the non-negativity of the parameters in the SDAE. Experimental results on TIMIT with 20 noise types at various noise levels demonstrate the superiority of the proposed method over the conventional baselines.**

***Index Terms*—Speech Enhancement, Unseen Noise Compensation, Deep Auto Encoder, Source Separation**

I. Introduction

Speech enhancement is an important stage to improve the perceptual quality of a noisy speech signal. The core problem in speech enhancement is the separation of speech and noise, for which a commonly deployed technique is estimating and removing the noise spectrum from the input noisy speech spectrum. If representative noise samples are available before conducting the enhancement, one can extract and exploit the spectral characteristics. However, an ideal technique should hold good performance in unseen noise conditions and not be limited to several known noise types.

Another difficulty concerning the research of speech enhancement is that many types of noises are non-stationary. In contrast to stationary noises, the spectral properties of non-stationary ones are difficult to predict and estimate, which makes noise removal challenging. Speech enhancement with seen noises is straightforward and will not be discussed in this paper. For the remaining cases, the related research is discussed as follows.

A. Stationary Noises Unseen in the Training Set

Spectral subtraction (SS) [1, 2] and minimum mean square error (MMSE) estimators [3–5] were proposed for speech enhancement in stationary noise environments. These methods do not require any prior knowledge about noise signals, nor any training stage beforehand, so they can work for unseen noise cases. Normally, these algorithms assume that the noise is stationary which is the foundation that the full noise spectrum can be predicted by only using the non-speech intervals decided by voice activity detection (VAD). However, the performance of VAD largely depends on the noise level and the noise type. It can fail in the presence of strong non-stationary noises [6]. Therefore, we will not consider these algorithms in the current context.

B. Non-Stationary Noises Seen in the Training Set

Hidden Markov models [7] and codebooks of linear prediction coefficients (LPC) [8] were introduced to model non-stationary noises for speech enhancement. In [7], speech and non-stationary noise were modeled by two different HMMs whose states were coupled like in factorial HMMs [9]. In [8], codebooks were trained from noise data as prior knowledge. However, the methods are based on the assumption that the training noises hold the same spectral characteristics as the testing noises, so it will fail to cope with unseen noise types, especially when training and testing noises behave differently.

In a recent work on deep auto encoder (DAE)[10], stereo training data were created by artificially adding noises to clean speech samples and training the DAE with the noisy data as input and the clean data as output. The experiment was done using two non-stationary noise types (*car* and *factory*) but evaluation was constrained to the same noise types. Hence, generalization to unseen noise types was not shown.

Meng Sun and Xiongwei Zhang are with the Lab of Intelligent Information Processing, PLA University of Science and Technology, 210007 Nanjing, China. E-mail: sunmengccjs@gmail.com.

Hugo Van hamme is with the Speech Processing Research Group, Electrical Engineering Department (ESAT), KU Leuven, 3000 Leuven, Belgium.

Thomas Fang Zheng is with the Research Institute of Information Technology, Tsinghua University, Beijing 100084, China.

The research of M. Sun and X. Zhang was funded by the Natural Science Foundation of China (61471394, 61402519) and the Natural Science Foundation of Jiangsu Province (BK20140071, BK20140074, BK2012510). The research of H. Van hamme was funded by the KULeuven research grant GOA/14/005 (CAMETRON). The research of T. F. Zheng was supported by the National Natural Science Foundation of China under Grant No. 61271389 and the National Basic Research Program (973 Program) of China under Grant No. 2013CB329302.

C. Non-Stationary Unseen Noises With Low-Rank Structures

To separate speech and noise in the spectral domain, nonnegative dictionary learning has been extensively studied recently [11]. In this idea, one first trains two groups of nonnegative bases: the speech related basis and the noise related basis. The noisy input speech spectrum is subsequently represented by the convex combination of both bases. Finally, the clean speech spectrum is estimated by the linear combination of the speech bases weighted by their coefficients.

In case of an unseen noise type, no noise dictionary can be obtained beforehand. Hence, only a speech dictionary is available and leaving the noise bases to be learned on the fly during the enhancement as presented in [12] and [13]. Another approach is to train a group of noise bases from some known noise types and then to utilize the bases to unseen noise conditions, regardless of the possible mismatch between the training and testing noises as reported in [14].

In the technique where noise bases are learned on the fly, the noise spectrogram is assumed to be low rank, i.e. containing a couple of *repeated* spectral structures. Since no assumption about the stationary property is required, the method is able to model transient noise types [12]. However, for non-stationary noises without repeated low rank spectral structures, the method might fail. In this paper, we take this method as a baseline to evaluate our proposed approach.

D. Non-Stationary Unseen Noises Predictable by Hidden Markov Models

To estimate the spectral characteristics of unseen non-stationary noises, stochastic gain HMM (SGHMM) was investigated in [15] for online noise estimation by updating the auto-regression HMM (ARHMM) parameters of the noise in a recursive EM framework. The ARHMM was utilized to model the linear predictive coefficients (LPC) by setting a special covariance matrix for the Gaussian distribution of each HMM state. The stochastic gain was employed to tune the energy fluctuations caused by the changing of distance from the speaker to the microphone.

To accommodate the unseen non-stationary noise power spectrum, an adaptive HMM was proposed in [16] by designing a functional as the observation probability density for each HMM state. Spectral domain speech enhancement using HMM state-dependent super-Gaussian priors was proposed in [17]. Those methods are based on the assumption that the unseen noise spectrum is predictable by HMMs, i.e. that the dynamic nature of the non-stationary noises can be captured by state transitions. Our proposed method will also be compared against this class of methods.

E. Recent Efforts on Modeling Non-Stationary Unseen Noises Using Deep Learning

In [18], 104 noise types were utilized to synthesize training data of noisy speech. Deep neural networks (DNNs) were trained as a mapping function from noisy to clean speech signals in a similar way as presented in the second paragraph of section I-B. The large training set encompasses many possible combinations of speech and noise types. Hence, the learned DNNs were expected to handle unseen noises in real-world situations. Experimental studies were conducted and significant improvements were observed in [18]. With sufficient training samples of noises, the method appeared to work for many unseen case. However, it seems impossible to make a universal noise data set to cover all unseen noise types. Therefore, alternative strategies make sense besides the approaches using big data.

F. The Motivation of This Work

In this work, we investigate a method without any pre-training of noise models. The only assumption about the noise is that it is *different* from the involved speech. Therefore, the noise estimation turns out to be finding the components which cannot be adequately represented by a well defined speech model. Given the good performance of deep learning in signal representation, a deep auto encoder (DAE) is employed for accurately modeling clean speech spectrum, whose configuration details are given in section III-A. In the enhancement stage, an extra DAE is introduced to represent the residual part obtained by subtracting the estimated clean speech spectrum (by using the pre-trained DAE) from the noisy speech spectrum, as presented in section II-A. By adjusting the estimated clean speech spectrum and the unknown parameters of the noise DAE, one can reach a stationary point to minimize the total reconstruction error of the noisy speech spectrum. Enhanced speech is then obtained by transforming the estimated clean speech spectrum back into time domain. Meanwhile, the noise and its spectrum can also be obtained as a by-product. The above proposed technique is called separable deep auto encoder (SDAE) since it contains two parallel parts: a pre-trained DAE to represent signals from one source and a DAE trained on the fly to represent signals from the other source(s). Given the under-determined nature of the above SDAE's optimization problem, the clean speech reconstruction is confined in the convex hull spanned by a pre-trained speech dictionary in section II-B. In [19], the authors proposed a nonnegative deep network architecture results from unfolding the iterations and untying the parameters of NMF. This architecture retained the basis additivity assumption of NMF and was believed to have more powerful representation ability than NMF. To optimize its nonnegative parameters, the authors derived multiplicative back propagation updating rules which can be used to preserve nonnegativity without the need for constrained optimization. In our proposed deep learning neural network, nonnegativity is also expected for the basis activation coefficients of the speech DAE and the weighting and bias parameters of the noise DAE. Inspired by their multiplicative back propagation in [19], we solve our problem in a conceptually similar way but with different updating rules to optimize a different objective function. New learning algorithms are investigated to respect the non-negativity of the parameters in SDAE in section II-B. Detailed configuration of the noise DAE is described in section III-B. Experimental results on TIMIT with 20 noise types at various noise levels will be reported and analyzed in section IV.

Compared with SS and MMSE in I-A, our method is able to cope with both stationary and non-stationary noises.

Compared with the HMM and codebook-driven approaches in I-B, our method does not assume the noise is known beforehand. Compared with the dictionary learning methods in I-C, the noises treated by our method are not limited to those with low-rank spectrum structures. Compared with the HMM methods in I-D, our method works in a frame-by-frame fashion, so no slow changes nor first-order state transition is imposed on the dynamic properties of non-stationary noises, which helps our method to model transient noises with abrupt changes. Compared with I-E, no pre-training of noise models is required, so the performance does not rely on the amount of noise training data.

II. Separable Deep Auto Encoder for Speech and Noise Separation

A. Separable Deep Auto Encoder: A General Framework

Let $\mathbf{X}$ denote the spectrogram of the input noisy speech, and let $\hat{\mathbf{X}}$, $\hat{\mathbf{S}}$ and $\hat{\mathbf{N}}$ denote the reconstructed spectrograms of noisy speech, clean speech and noise, respectively. $\hat{\mathbf{x}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{n}}$ represent the spectrum of one particular frame. With the approximation that the spectra are additive, we have,

$$\begin{aligned}\hat{\mathbf{x}} &= \hat{\mathbf{s}} + \hat{\mathbf{n}}, \\ &= f(\mathbf{s}) + g(\mathbf{n}). \end{aligned} \tag{1}$$

In (1), $\hat{\mathbf{s}}$ and $\hat{\mathbf{n}}$ have been represented by functions of their oracle counterparts $\mathbf{s}$ and $\mathbf{n}$, denoted by $f(\mathbf{s})$ and $g(\mathbf{n})$. Two deep auto encoders (DAE) are utilized to represent $f(\mathbf{s})$ and $g(\mathbf{n})$, respectively:

$$f(\mathbf{s}) = \sigma(\mathbf{W}^{(2)}\sigma(\mathbf{W}^{(1)}\mathbf{s} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}), \tag{2}$$

$$g(\mathbf{n}) = \sigma(\mathbf{V}^{(2)}\sigma(\mathbf{V}^{(1)}\mathbf{n} + \mathbf{c}^{(1)}) + \mathbf{c}^{(2)}). \tag{3}$$

Without loss of generality and for simplicity, the number of layers is chosen as 2 as an example here. $\sigma(.)$ is the activation function, for which a Rectified Linear Unit (ReLU) [20] is selected in this work to ensure the nonnegativity of the reconstructed spectrum. The mathematical definition of ReLU is $\sigma(\cdot) = \max(\cdot, 0)$.

In the task of speech enhancement, we assume that a deep auto encoder for modeling any clean speech is already available, whose nodes are denoted by the filled circles in Figure 1. That is $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ have been learned from some training data of clean speech and they act as prior knowledge for the forthcoming speech enhancement. The reader is referred to section III-A for the details. Hence, the unknown parameters to be estimated are: the clean speech spectrum $\mathbf{s}$, the noise spectrum $\mathbf{n}$, the weighting terms $\mathbf{V}^{(k)}$ and the bias terms $\mathbf{c}^{(k)}$ in noise DAE. In light of the idea of spectral subtraction, the noise spectrum $\mathbf{n}$ may be replaced by $\sigma(\mathbf{x} - \mathbf{s})$, where the ReLU function $\sigma(.)$ is again imposed to retain the nonnegativity of the noise spectrum. Hereby, the reconstruction formula of noise is thus converted into a function of $\mathbf{s}$[1],

$$g(\mathbf{s}) = \sigma(\mathbf{V}^{(2)}\sigma(\mathbf{V}^{(1)}(\sigma(\mathbf{x} - \mathbf{s})) + \mathbf{c}^{(1)}) + \mathbf{c}^{(2)}). \tag{4}$$

[1]For simplicity but without bringing confusion, we did not change the symbol g to represent a different function from (3).

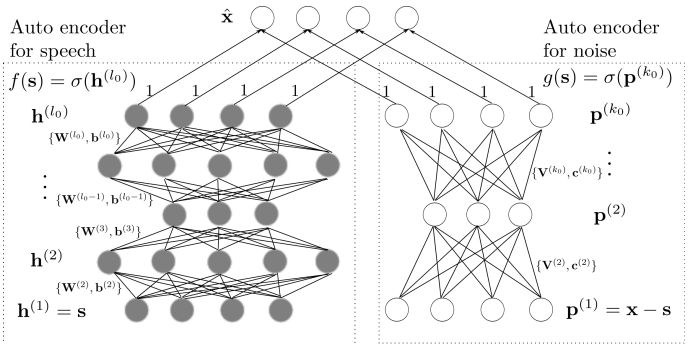

Fig. 1. The architecture of the proposed separable deep auto encoder. The neural network contains two parts: a pre-trained deep auto encoder to represent clean speech denoted by filled circles and a DAE trained on the fly to represent noise denoted by blank circles. The sum of the outputs of the top layers of the two DAEs is expected to be the noisy observation, i.e. $\mathbf{x} \approx \hat{\mathbf{x}} = f(\mathbf{s}) + g(\mathbf{s})$. The bottom layers are the unknown clean speech spectrum $\mathbf{s}$ and the estimated noise spectrum $\mathbf{x} - \mathbf{s}$.

As illustrated in Figure 1, the sum of the outputs of the top layers of the speech and noise DAEs is expected to be close to the noisy observation, i.e. $\mathbf{x} \approx \hat{\mathbf{x}} = f(\mathbf{s}) + g(\mathbf{s})$. The optimization problem thus boils down to the following configuration,

$$\{\hat{\mathbf{s}}, \hat{\mathbf{V}}^{(k)}, \hat{\mathbf{c}}^{(k)}\} = \underset{\mathbf{s}, \mathbf{V}^{(k)}, \mathbf{c}^{(k)}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{x} - (f(\mathbf{s}) + g(\mathbf{s}))||_2^2, \tag{5}$$

with the constraint that the entries in $\mathbf{s}$ are all nonnegative and smaller than their counterparts in $\mathbf{x}$. In (5), we have opted for a Euclidean cost function. Given its success in source separation on magnitude spectra [21], we have also considered Kullback-Leibler divergence. However, so far we have not been able to obtain superior results with this cost function.

The above problem is under-determined and additional constraints should be imposed. We hereby introduce a speech dictionary to do so.

B. Confining the Speech Spectrogram in the Convex Hull Spanned by Nonnegative Speech Bases

Given the success of dictionary learning in speech enhancement, we represent the speech spectrogram by the convex combination of nonnegative speech bases in a speech dictionary $\mathbf{D}$. The dictionary is usually learned from a large amount of clean speech data. Hence, $\mathbf{s}$ in (2) and (4) can be replaced by $\mathbf{D}\mathbf{y}$ where $\mathbf{y}$ is the coefficient vector of the speech bases. The reconstruction formulae of speech and noise are thus converted into the functions of $\mathbf{y}$,

$$f(\mathbf{y}) = \sigma(\mathbf{W}^{(2)}\sigma(\mathbf{W}^{(1)}(\mathbf{D}\mathbf{y}) + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}), \tag{6}$$

$$g(\mathbf{y}) = \sigma(\mathbf{V}^{(2)}\sigma(\mathbf{V}^{(1)}(\sigma(\mathbf{x} - \mathbf{D}\mathbf{y})) + \mathbf{c}^{(1)}) + \mathbf{c}^{(2)}). \tag{7}$$

Therefore, the optimization problem now becomes,

$$\{\hat{\mathbf{y}}, \hat{\mathbf{V}}^{(k)}, \hat{\mathbf{c}}^{(k)}\} = \underset{\mathbf{y}, \mathbf{V}^{(k)}, \mathbf{c}^{(k)}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{x} - (f(\mathbf{y}) + g(\mathbf{y}))||_2^2, \tag{8}$$

with the constraint that the elements in $\mathbf{y}$ are nonnegative. To respect the nonnegativity of the involved parameters and

inspired by [19], we take the multiplicative updating rules for $\mathbf{V}^{(k)}$, $\mathbf{c}^{(k)}$ and $\mathbf{y}$ as follows,

$$\mathbf{y} \leftarrow \mathbf{y} \odot \left(\left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^- \oslash \left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^+ \right)^{\gamma}, \tag{9}$$

$$\mathbf{V}^{(k)} \leftarrow \mathbf{V}^{(k)} \odot \left(\left[\frac{\partial \epsilon}{\partial \mathbf{V}^{(k)}}\right]^- \oslash \left[\frac{\partial \epsilon}{\partial \mathbf{V}^{(k)}}\right]^+ \right)^{\gamma}, \tag{10}$$

$$\mathbf{c}^{(k)} \leftarrow \mathbf{c}^{(k)} \odot \left(\left[\frac{\partial \epsilon}{\partial \mathbf{c}^{(k)}}\right]^- \oslash \left[\frac{\partial \epsilon}{\partial \mathbf{c}^{(k)}}\right]^+ \right)^{\gamma}, \tag{11}$$

where

$$\epsilon = \frac{1}{2}||\mathbf{x} - (f(\mathbf{y}) + g(\mathbf{y}))||_2^2$$

is the reconstruction error, and $\odot$ and $\oslash$ are the element-wise multiplication and division, respectively. γ is the tunable exponential step-size, whose initial value is 1. Once the non-decreasing of the cost function is observed, γ will be reduced to its half in the next iteration.

The chain rule is subsequently utilized to derive the positive and negative parts of the gradients for (9)$\sim$(11). It is worth to clarify that the vectors in this paper all refer to column vectors.

1) Derivation of the Updating Rules for $\mathbf{y}$*:* For the coefficients vector $\mathbf{y}$, we have,

$$\left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^+ = \left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^+ \left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^+ + \left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^- \left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^- \tag{12}$$

$$\left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^- = \left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^- \left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^+ + \left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^+ \left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^- \tag{13}$$

Since $\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}} = \mathbf{D}^T$ and $D_{i,j} \geq 0$, we have $\left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^+ = \mathbf{D}^T$ and $\left[\frac{\partial \mathbf{s}^T}{\partial \mathbf{y}}\right]^- = \mathbf{0}$. To obtain the positive and negative parts of $\frac{\partial \epsilon}{\partial \mathbf{s}}$ in a recursive way, we first introduce the layer-wise notation of the speech DAE,

$$\mathbf{h}^{(l+1)} = \mathbf{W}^{(l+1)}\sigma(\mathbf{h}^{(l)}) + \mathbf{b}^{(l+1)}$$

where $1 \leq l \leq l_0 - 1$ is the layer index, $\mathbf{h}^{(1)} = \mathbf{s}$ and $\sigma(\mathbf{h}^{(l_0)}) = f(\mathbf{y})$. Similarly, for the noise part the layer-wise notation is,

$$\mathbf{p}^{(k+1)} = \mathbf{V}^{(l+1)}\sigma(\mathbf{p}^{(l)}) + \mathbf{c}^{(l+1)}$$

where $1 \leq k \leq k_0 - 1$ is the layer index, $\mathbf{p}^{(1)} = \mathbf{x} - \mathbf{s}$ and $\sigma(\mathbf{p}^{(k_0)}) = g(\mathbf{y})$. Given the definition of ϵ and (2) and (4), the partial derivatives of ϵ with respect to $\mathbf{s}$ is derived as follows,

$$\frac{\partial \epsilon}{\partial \mathbf{s}} = \frac{\partial (\sigma(\mathbf{h}^{(1)}))^T}{\partial \mathbf{s}} \frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(1)})} + \frac{\partial (\sigma(\mathbf{p}^{(1)}))^T}{\partial \mathbf{s}} \frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(1)})} \tag{14}$$

Given the above definitions of $\mathbf{h}^{(1)}$ and $\mathbf{p}^{(1)}$, we can obtain the derivatives $\frac{\partial (\sigma(\mathbf{h}^{(1)}))^T}{\partial \mathbf{s}} = \mathbf{I}$ and $\frac{\partial (\sigma(\mathbf{p}^{(1)}))^T}{\partial \mathbf{s}} = -\text{sign}(\mathbf{x}-\mathbf{s})$ where $\mathbf{I}$ is the identity matrix and $\text{sign}(\cdot)$ is the sign function. By using a similar splitting trick presented in (12) and (13), the positive and negative parts of $\frac{\partial \epsilon}{\partial \mathbf{s}}$ will subsequently be obtained. Hereby, the remaining problem is to derive the bottom-up recursive rules which are given as follows,

$$\begin{aligned}
&\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l)})}\right]^+ \\
&= \left[\frac{\partial (\mathbf{h}^{(l+1)})^T}{\partial \sigma(\mathbf{h}^{(l)})}\right]^+ \left[\frac{\partial (\sigma(\mathbf{h}^{(l+1)}))^T}{\partial \mathbf{h}^{(l+1)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^+ \\
&\quad + \left[\frac{\partial (\mathbf{h}^{(l+1)})^T}{\partial \sigma(\mathbf{h}^{(l)})}\right]^- \left[\frac{\partial (\sigma(\mathbf{h}^{(l+1)}))^T}{\partial \mathbf{h}^{(l+1)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^-
\end{aligned} \tag{15}$$

$$\begin{aligned}
&= \left[\mathbf{W}^{(l+1)}\right]^{+,T} \text{diag}(\mathcal{I}(\mathbf{h}^{(l+1)})) \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^+ \\
&\quad + \left[\mathbf{W}^{(l+1)}\right]^{-,T} \text{diag}(\mathcal{I}(\mathbf{h}^{(l+1)})) \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^-,
\end{aligned} \tag{16}$$

$$\begin{aligned}
&\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l)})}\right]^- \\
&= \left[\frac{\partial (\mathbf{h}^{(l+1)})^T}{\partial \sigma(\mathbf{h}^{(l)})}\right]^- \left[\frac{\partial (\sigma(\mathbf{h}^{(l+1)}))^T}{\partial \mathbf{h}^{(l+1)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^+ \\
&\quad + \left[\frac{\partial (\mathbf{h}^{(l+1)})^T}{\partial \sigma(\mathbf{h}^{(l)})}\right]^+ \left[\frac{\partial (\sigma(\mathbf{h}^{(l+1)}))^T}{\partial \mathbf{h}^{(l+1)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^-
\end{aligned} \tag{17}$$

$$\begin{aligned}
&= \left[\mathbf{W}^{(l+1)}\right]^{-,T} \text{diag}(\mathcal{I}(\mathbf{h}^{(l+1)})) \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^+ \\
&\quad + \left[\mathbf{W}^{(l+1)}\right]^{+,T} \text{diag}(\mathcal{I}(\mathbf{h}^{(l+1)})) \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l+1)})}\right]^-
\end{aligned} \tag{18}$$

for layer l from 1 to $l_0 - 1$. The middle term $\left[\frac{\partial (\sigma(\mathbf{h}^{(l+1)}))^T}{\partial \mathbf{h}^{(l+1)}}\right]$ in the above chain rules is the derivative of the ReLU function which is always nonnegative. In this context, it is a diagonal matrix whose diagonal elements are the indicator values whether $h_i^{(l+1)}$ is positive or not ($\mathcal{I}(\cdot)$ is the indicator function). Hence, it has no impact on the signs of other terms.

At the end of the above recursive steps, i.e. when reaching the top layer l_0 of the speech DAE, we have,

$$\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^+ = \sigma(\mathbf{h}^{(l_0)}) + \sigma(\mathbf{p}^{(k_0)}), \tag{19}$$

$$\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^- = \mathbf{x}. \tag{20}$$

To compute the second part in (14), $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^-$ can be derived in a similar way as above. However, we should note that the above derivation process involves computing the positive and negative parts of the $\mathbf{V}^{(k)}$'s which are unknown variables to be estimated, unlike the $\mathbf{W}^{(l)}$ which are kept fixed. Thanks to the nonnegativity constraints on the elements of each $\mathbf{V}^{(k)}$, its positive part is always itself and its negative part is zero. The nonnegativity constraints on $\mathbf{V}^{(k)}$ and $\mathbf{c}^{(k)}$ simplify the multiplicative updating of $\mathbf{y}$ at the expense that ReLU makes no sense for the layers from 2 to k_0 in the noise DAE.

2) Derivation of the Updating Rules for the Parameters in Noise DAE: To derive $\frac{\partial \epsilon}{\partial \mathbf{V}^{(k)}}$ and $\frac{\partial \epsilon}{\partial \mathbf{c}^{(k)}}$, we first need to obtain the derivatives of ϵ with respect to the noise representation at layer k, $\sigma(\mathbf{p}^{(k)})$, i.e. $\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}$. It should be conducted in a top-down recursive fashion. At the start of the recursion, the derivatives of ϵ with respect to the top-layer units of noise are,

$$\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^+ = \sigma(\mathbf{h}^{(l_0)}) + \sigma(\mathbf{p}^{(k_0)}), \tag{21}$$

$$\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^- = \mathbf{x}. \tag{22}$$

With similar rules as described in (15)$\sim$(18), we can obtain $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^-$ for any k from $k_0 - 1$ to 2 by just replacing $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ with $\mathbf{V}^{(k)}$ and $\mathbf{c}^{(k)}$, respectively. Noting the nonnegativity of $\mathbf{V}^{(k)}$, its negative part is always zero. Hence, the second term of (16) and the first term of (18) can be removed. Finally, the gradients of the reconstruction error ϵ with respect to the noise DAE parameters are,

$$\begin{aligned}
&\left[\frac{\partial \epsilon}{\partial V_{i,j}^{(k)}}\right]^+ \\
=&\left[\frac{\partial (\mathbf{p}^{(k)})^T}{\partial V_{i,j}^{(k)}}\right]^+ \left[\frac{\partial (\sigma(\mathbf{p}^{(k)}))^T}{\partial \mathbf{p}^{(k)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^+ \\
&+ \left[\frac{\partial (\mathbf{p}^{(k)})^T}{\partial V_{i,j}^{(k)}}\right]^- \left[\frac{\partial (\sigma(\mathbf{p}^{(k)}))^T}{\partial \mathbf{p}^{(k)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^- \\
=&\left[\sigma(\mathbf{p}^{(k-1)})\right]^{+,T} \left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^+ + \left[\sigma(\mathbf{p}^{(k-1)})\right]^{-,T} \left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^- \\
=&(\mathbf{p}^{(k-1)})^T \left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^+,
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\left[\frac{\partial \epsilon}{\partial c_i^{(k)}}\right]^+ \\
=&\left[\frac{\partial (\mathbf{p}^{(k)})^T}{\partial c_i^{(k)}}\right]^+ \left[\frac{\partial (\sigma(\mathbf{p}^{(k)}))^T}{\partial \mathbf{p}^{(k)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^+ \\
&+ \left[\frac{\partial (\mathbf{p}^{(k)})^T}{\partial c_i^{(k)}}\right]^- \left[\frac{\partial (\sigma(\mathbf{p}^{(k)}))^T}{\partial \mathbf{p}^{(k)}}\right] \left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^- \\
=&\mathbf{1}^T \left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^+
\end{aligned} \tag{24}$$

In the above derivations, we have utilized the fact that $\sigma(\mathbf{p}^{(k)}) = \left[\mathbf{p}^{(k)}\right]^+ = \mathbf{p}^{(k)}$ and $\left[\mathbf{p}^{(k)}\right]^- = \mathbf{0}$, for any integer $k \in [2, k_0]$. Specifically, $\left[\sigma(\mathbf{p}^{(1)})\right]^+ = \sigma(\mathbf{x} - \mathbf{D}\mathbf{y})$.

The negative parts $\left[\frac{\partial \epsilon}{\partial V_{i,j}^{(k)}}\right]^-$ and $\left[\frac{\partial \epsilon}{\partial c_i^{(k)}}\right]^-$ are straightforwardly obtained by replacing $\left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^+$ with $\left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^-$ in (23) and (24), respectively.

III. Model Training and Tuning

A. Clean Speech Magnitude Spectrogram Reconstruction

In this section, we present how a DAE was trained for clean speech reconstruction on 500 utterances randomly selected from the TIMIT dataset. In our work, the magnitude spectrum was extracted for each frame and was chosen as the input features to the neural network. ReLU was chosen as the activation function to maintain the nonnegativity of the spectrum but without compressing the amplitudes of the spectra. The objective function was to minimize the Euclidean distance between the input spectrum and its reconstruction using DAE.

A 512-point FFT was firstly conducted for each windowed frame to result in a 257 dimensional spectrum vector. For all the 500 utterances, this yielded around 200k vectors for training. Subsequently, layer-wise unsupervised pre-training was deployed in a greedy way to construct the deep architecture. 1000 hidden units were learned by training an auto encoder with the structure 257×1000×257. To make the representation deep, the 1000 hidden units were further encoded by 200 hidden units by training a second auto encoder with the structure 1000×200×1000. Finally, by unfolding and stacking the two auto encoders, we obtained a 5-layer DAE with size 257×1000×200×1000×257, i.e. by replacing the 1000 hidden units in the middle layer of the first network (with structure 257×1000×257) with the whole second network (with structure 1000×200×1000). After the above pre-training, supervised fine tuning with back propagation was conducted to refine the parameters.

The optimization strategy in the above pre-training and fine tuning was stochastic gradient descent where each batch contained 1000 frames and the number of iterations was set 1000. No sparsity penalty was imposed to the cost function. A decreasing learning rate was adopted to ensure the convergence where the learning rate was reduced to its half once the cost value was observed increasing. The initial learning rate was 0.1. The momentum was chosen as 0.1. To avoid numerical overflow, the gradients were normalized to hold the unity ℓ^∞ norm, e.g.

$$\frac{\partial \epsilon}{\partial \mathbf{W}^{(l)}} \leftarrow \frac{\partial \epsilon}{\partial \mathbf{W}^{(l)}} / || \frac{\partial \epsilon}{\partial \mathbf{W}^{(l)}} ||_\infty. \tag{25}$$

B. Noise modeling

1) Network Configuration and Initialization: For noise modeling, we configured a 3-layer DAE with size $257 \times M \times 257$ where M is the number of hidden units. As presented in section II-A, the noise DAE was learned per utterance. Hence, the number of parameters might be determined by the length of the utterance and by the amounts of variations of the noise spectrogram. Generally, for long utterances with strong noises, more units should be introduced. Parameter sensitivity of the number of units M will be illustrated in the figures of section IV-E, IV-F and IV-H.

The noise DAE interacts with the speech DAE model in two aspects: one is sharing the same cost function ϵ at the top layer and the other is acquiring its inputs by subtracting the speech spectrums from the noisy inputs at the bottom layer, i.e. $\sigma(\mathbf{x} - \mathbf{s})$. These two aspects provide the foundation of joint training of the parameters of speech and noise.

The initial values of the entries in $\mathbf{V}^{(k)}$ and $\mathbf{c}^{(k)}$ were all set to 1 which has no scaling impact on the subsequent multiplicative updates as presented in (10) and (11).

2) Large Margin Constraints: Given the speech and noise reconstructions, $f(\mathbf{s})$ and $g(\mathbf{s})$ (i.e. $\sigma(\mathbf{h}^{(l_0)})$ and $\sigma(\mathbf{p}^{(k_0)})$ in their multi-layer representations, respectively), a regularization term was added to the original cost function,

$$\begin{aligned}\mathcal{R}_1 &= -\alpha||f(\mathbf{s}) - g(\mathbf{s})||_2^2 \\ &= -\alpha||\sigma(\mathbf{h}^{(l_0)}) - \sigma(\mathbf{p}^{(k_0)})||_2^2 \end{aligned} \tag{26}$$

The above equation tends to increase the dissimilarity between the two sources by suppressing signals from other sources in the current source prediction [22]. Therefore, the above regularization term makes the learning problem discriminative. In our experiments, a small α would work while a big one hindered the convergence of the proposed algorithm. This is because the concave part can make the gradient small close to a maximum. On the other hand, without using the regularization, the performance deteriorated a bit. In this paper, α was set as 0.1. The parameter sensitivity regarding this parameter will be discussed in section IV-H.

The positive and negative part of the gradient of $\mathcal{R}_1$ with respect to $\sigma(\mathbf{h}^{(l_0)})$ and $\sigma(\mathbf{p}^{(k_0)})$ are as follows,

$$\left[\frac{\partial \mathcal{R}_1}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^+ = 2\alpha\sigma(\mathbf{p}^{(k_0)}) \tag{27}$$

$$\left[\frac{\partial \mathcal{R}_1}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^- = 2\alpha\sigma(\mathbf{h}^{(l_0)}) \tag{28}$$

$$\left[\frac{\partial \mathcal{R}_1}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^+ = 2\alpha\sigma(\mathbf{h}^{(l_0)}) \tag{29}$$

$$\left[\frac{\partial \mathcal{R}_1}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^- = 2\alpha\sigma(\mathbf{p}^{(k_0)}) \tag{30}$$

3) Dropout of the Noise Units to Retain Sparsity: A method called "dropout" was proposed in [23] to improve the generalization capability of deep neural networks to avoid over fitting. In this technique, dropout randomly omits a certain percentage of the units in the input and each hidden layer during each presentation of every training sample. In this paper, we implemented this idea to noise modeling to retain sparsity. For each frame, only a small portion (say η) among the total M units were activated. To choose the active units, we first sorted the weights of all the units in a decreasing order, then selected the top activated ones. The sensitivity of the model's performance with respect to η will be reported later in section IV-H.

C. An Overview

The overall flowchart is given in Figure 2. It contains two parts: the "offline training" and the "online enhancement". In the "offline training" part, given a collection of clean speech spectrum $\mathbf{S}$, a nonnegative dictionary $\mathbf{D}$ was first learned by using NMF. A deep auto encoder $f(\mathbf{s})$ was trained to reconstruct $\mathbf{S}$ from itself as presented in section III-A.

In the "online enhancement" state, we first extract the spectrogram of the noisy utterance, then construct and train a separable DAE to estimate the clean speech spectrum as well as the noise spectrum. The algorithm is summarized in Algorithm 1. Finally, clean speech signal is reconstructed by using the estimated spectrum for clean speech and the phase from the noisy input.

Algorithm 1 The learning algorithm of SDAE

Input: $\mathbf{x}, \mathbf{D}, \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}, l = 1, \ldots, l_0\}$
Output: $\hat{\mathbf{s}}, \{\mathbf{V}^{(k)}, \mathbf{c}^{(k)}, k = 1, \ldots, k_0\}$

1: t=1,$\delta = +\infty$,initialize $\mathbf{y}$, $\mathbf{V}^{(k)}$ and $\mathbf{c}^{(k)}$
2: **while** $t \leq T$ && $\delta >$ threshold **do**
3: //Feed forward of each neural network
4: $\mathbf{h}^{(1)} = \mathbf{D}\mathbf{y}$
5: **for** $l = 2; l \leq l_0; l++$ **do**
6: Compute $\mathbf{h}^{(l)} = \mathbf{W}^{(l)}\sigma(\mathbf{h}^{(l-1)}) + \mathbf{b}^{(l)}$
7: **end for**
8: $\mathbf{p}^{(1)} = \mathbf{x} - \mathbf{D}\mathbf{y}$
9: **for** $k = 2; k \leq k_0; k++$ **do**
10: Compute $\mathbf{p}^{(k)} = \mathbf{V}^{(k)}\sigma(\mathbf{p}^{(k-1)}) + \mathbf{c}^{(k)}$
11: **end for**
12: $\tilde{\delta} = \frac{1}{2}||\mathbf{x} - \sigma(\mathbf{h}^{(l_0)}) - \sigma(\mathbf{p}^{(k_0)})||_2^2 - \alpha||\sigma(\mathbf{h}^{(l_0)}) - \sigma(\mathbf{p}^{(k_0)})||_2^2$
13: **if** $\tilde{\delta} >= \delta$ **then**
14: $\gamma \leftarrow \gamma/2$
15: **end if**
16: $\delta = \tilde{\delta}$
17: //Compute the gradients using back propagation
18: Compute $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l_0)})}\right]^-$ by using (19)+(27) and (20)+(28), respectively
19: **for** $l = l_0 - 1; l \geq 1; l--$ **do**
20: Compute $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(l)})}\right]^-$ by using (16) and (18), recursively
21: **end for**
22: Compute $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^-$ by using (21)+(29) and (22)+(30), respectively
23: **for** $k = k_0 - 1; k \geq 1; k--$ **do**
24: Compute $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k)})}\right]^-$ recursively by replacing the speech DAE's variables in (16) and (18) with the noise DAE's ones
25: **end for**
26: With the outputs $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(1)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{h}^{(1)})}\right]^-$ from Line 21 and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \sigma(\mathbf{p}^{(k_0)})}\right]^-$ from Line 25, compute $\left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \mathbf{s}}\right]^-$ by splitting (14) into positive and negative parts
27: Compute $\left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial \mathbf{y}}\right]^-$ by applying (12) and (13)
28: **for** $k = k_0; k \geq 2; k--$ **do**
29: Compute $\left[\frac{\partial \epsilon}{\partial V_{i,j}^{(k)}}\right]^+$ and $\left[\frac{\partial \epsilon}{\partial c_i^{(k)}}\right]^+$ by using (23) and (24), respectively
30: Compute $\left[\frac{\partial \epsilon}{\partial V_{i,j}^{(k)}}\right]^-$ and $\left[\frac{\partial \epsilon}{\partial c_i^{(k)}}\right]^-$ by replacing $\left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^+$ with $\left[\frac{\partial \epsilon}{\partial \mathbf{p}^{(k)}}\right]^-$ in (23) and (24)
31: **end for**
32: //Update the parameters
33: Update $\mathbf{y}, \mathbf{V}^{(k)}, \mathbf{c}^{(k)}$ by using (9), (10) and (11), respectively
34: $t \leftarrow t + 1$
35: **end while**
36: $\hat{\mathbf{s}} = \mathbf{h}^{(l_0)}$
37: $\hat{\mathbf{n}} = \sigma(\mathbf{x} - \hat{\mathbf{s}})$

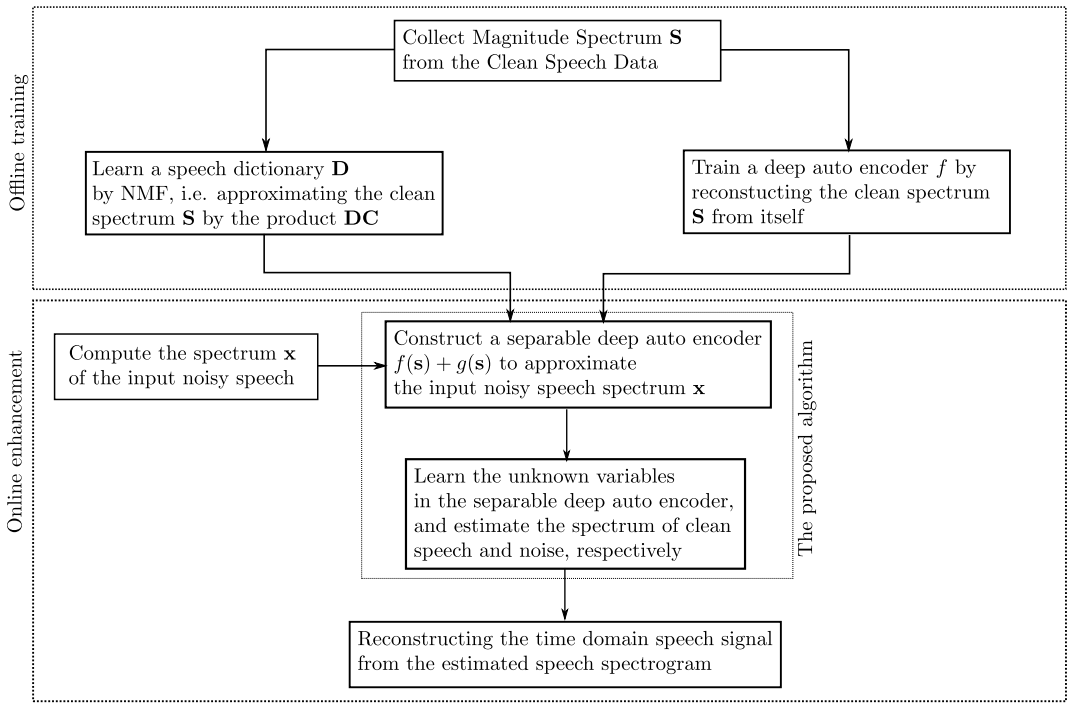

Fig. 2. The flowchart of the proposed method. The method contains two parts: the "offline training" and the "online enhancement". In the "offline training" part, given a collection of clean speech spectrum $\mathbf{S}$, a nonnegative dictionary $\mathbf{D}$ was first learned by using NMF. A deep auto encoder $f(\mathbf{s})$ was trained to reconstruct $\mathbf{S}$ from itself. In the "online enhancement" stage, we first extract the spectrogram of the noisy utterance, then construct and train a separable DAE to estimate the clean speech spectrum as well as the noise spectrum. Finally, clean speech signal is reconstructed by using the estimated spectrum for clean speech and the phase from the noisy input.

IV. Experiments and Results

A. Preparation of the Dataset

The proposed algorithms were evaluated with 100 noisy speech examples from male and female speakers, which were synthesized by adding clean speech to a variety of noise signals at different SNRs. Clean speech examples were chosen from the TIMIT dataset randomly (without overlapping speakers with the training utterances in section III-A). For the noise samples, seventeen types of noise from the Noizeus-92 dataset, *babble, birds, casino, cicadas, computerkeyboard, eatingchips, f16, factory1, factory2, frogs, jungle, machineguns, motorcycles, ocean, pink, white*, and *volvo*, were considered. Three more non-stationary noise types were included: *formula1, freeway* and *phone* which were from the "Formula One" file[2] in [24], the "Highway traffic" file in [25] and the "phone ringing.wav" file in [26], respectively. In total, twenty types of noise were evaluated. The signals were mixed at 4 different signal-to-noise ratios (SNRs) from -5 to 10 dB spaced by 5 dB. All files were resampled to 8 kHz sampling rate. To calculate the spectrograms we used a window length of 64 ms (512 points) and a frame shift of 8 ms (64 points).

B. Evaluation Metrics

Three metrics were computed to evaluate the performance of the speech enhancement algorithms. The first criterion was the PESQ score which measures the subjective speech quality [27]. The second metric was the signal-to-distortion ratio (SDR) value of the enhanced speech calculated by BSS-EVAL [28] to show the impacts on noise separation and suppression of the algorithms. An ideal algorithm should suppress noises without bringing too much distortion to the enhanced speech. The third one was the segmental SNR (segSNR) of the enhanced speech measured by the *composite_se* package from [27]. For all metrics, a larger score indicates better performance.

[2]The sound was from the engine of a racing car.

C. Baselines

The improved versions of SS and MMSE (log-MMSE) in [29] were taken as baseline algorithms. Besides these, we compared our method with the nonnegative dictionary learning approach using nonnegative matrix factorization (NMF) defined as below,

$$\underset{\mathbf{D}^{(n)},\mathbf{Z}^{(n)},\mathbf{Z}^{(s)}}{\operatorname{argmin}} \quad \mathrm{KLD}\left(\mathbf{X}||[\mathbf{D}^{(n)}\ \mathbf{D}]\begin{bmatrix}\mathbf{Z}^{(n)}\\ \mathbf{Z}^{(s)}\end{bmatrix}\right) \tag{31}$$

$$s.t. \quad D^{(n)}_{f,r} \geq 0, \sum_f D^{(n)}_{f,r} = 1, \forall r, \tag{32}$$

$$Z^{(n)}_{r,t} \geq 0, Z^{(s)}_{r,t} \geq 0. \tag{33}$$

where KLD is the extended Kullback-Leibler divergence performing as the cost function [21], $\mathbf{D}$ is the same speech dictionary as mentioned in section II-B, $\mathbf{D}^{(n)}$ is the noise dictionary to be learned from the noisy speech spectrogram $\mathbf{X}$, and $\mathbf{Z}^{(s)}$ and $\mathbf{Z}^{(n)}$ are the coefficient matrix of the speech bases and the noise bases, respectively. The number of speech

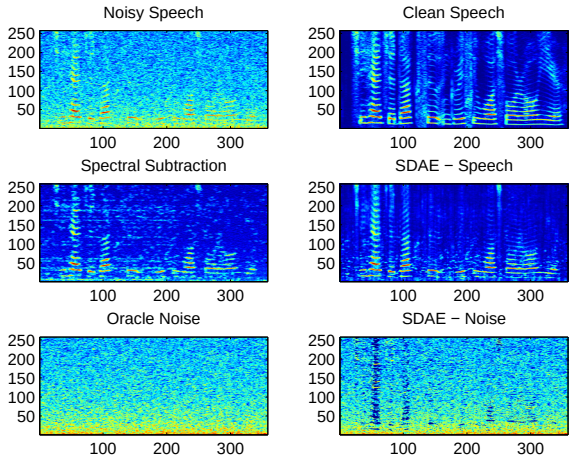

Fig. 3. The illustration of the performance of SDAE on *pink* noise at 0 dB SNR level. Better noise removing and few musical noise introducing, was observed in the output of SDAE than that from spectral subtraction. By contrasting the two bottom figures, one can see the good performance of noise estimation of SDAE.

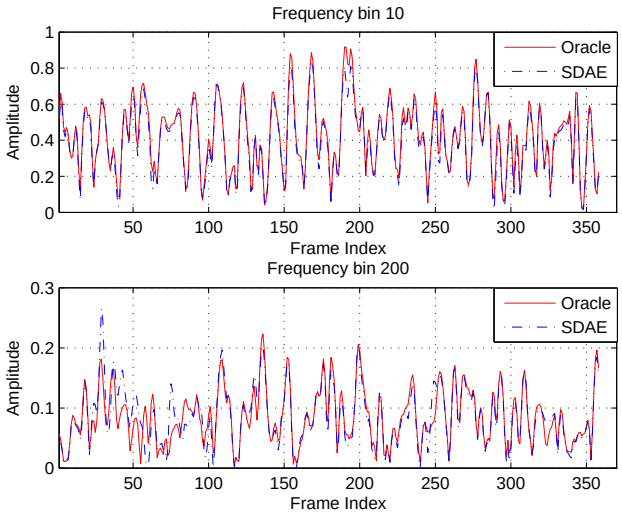

Fig. 4. Demonstration of the performance on noise spectrum estimation. The top figure shows the spectral envelopes at 156 Hz (corresponding to the 10th frequency bin). The bottom figure shows the spectral envelopes at 3125 Hz (corresponding to the 200th frequency bin). The red lines are the oracle/true values while the blue dashed lines are the results estimated by SDAE

bases, i.e. the number of columns in $\mathbf{D}$, was taken as 2000 for both the NMF baseline and the SDAE method with the NMF constraint in section II-B. The speech bases were learned by NMF from the training dataset as described in section III-A. The number of noise bases, i.e. the number of columns in $\mathbf{D}^{(n)}$, was chosen as 5. Experimental study showed more bases for noise modeling would also contain some speech structures and could deteriorate the model's performance, which may be due to the non-discriminative nature of the NMF.

Considering the HMM approaches presented in section I-D, we also compared our method with the performance of super-Gaussian HMM reported in [17] on three noise types (*babble*, *factory1* and *freeway*). The comparison with respect to the adaptive HMM proposed in [16] was also conducted on the two noise types (*car*+*phone* and *formula1*) in [16].

D. Visualization of the Noise Spectrum Estimation

Figure 3 visualizes the performance of the proposed method on *pink* noise at 0 dB SNR level. By comparing the bottom-left and the bottom-right figures, we can see that SDAE's good performance on noise spectrum estimation, except the "holes" generated by subtracting the speech spectrum components. In fact, the "holes" in the bottom-right figure are from the over-subtraction happened in $\sigma(\mathbf{x} - \mathbf{Dh})$. For the frequency bins where speech spectrum performs a dominant role, the entries in $\mathbf{Dh}$ might be larger than those in $\mathbf{x}$. To avoid negative values, the ReLU function $\sigma(.)$ forced them to zeros. However, this will not affect the performance on speech enhancement. The involved PESQ scores are: Noisy(1.49), SS(2.13), MMSE(2.12), NMF(2.30), and SDAE with NMF constraint(2.47), respectively.

Figure 4 shows the spectral envelopes at 150 Hz and 3125 Hz (corresponding to the 10th and 200th frequency bin, respectively). One can observe the good fit of the noise spectrum learned by DSAE compared to the oracle ones, especially for the low frequency case in the top figure of Figure 4.

E. The Method's General Performance

The PESQ scores, SDR values and segmental SNR values of the proposed methods as well as the SS/MMSE and NMF baselines are given in Figure 5, Figure 6 and Figure 7, respectively. "SS/MMSE" denotes the better performance among SS and MMSE for simplicity, a convention that is maintained throughout this text. From the figures, we see that both SDAE and NMF show significant improvements over SS/MMSE at low SNR levels. However, with increasing SNR, NMF deteriorates. SDAE demonstrates consistent and significant improvements over the baselines at all the four SNR levels by obtaining higher scores for PESQ, SDR and SEGSNR.

More units (i.e. larger M) in the noise DAE shows better performance at the low SNR level of -5 dB. Extensive discussion on the noise DAE's performance on the model's configuration will be presented in section IV-H.

F. The Model's Performance per Noise Type

To better understand the methods' performance on each noise type, Figure 8, Figure 9 and Figure 10 present the mean PESQ scores, SDR values and SEGSNR values over the four SNR levels, respectively. For the PESQ scores, from Figure 8, we can see SDAE outperforms SS/MMSE and NMF on most noise types, except the structural transient ones, like *machinegun*, *computerkeyboard*, *eatingchips* and *phone*. These noises hold low rank repeated structures which are particularly suited for modeling by NMF as also explored in [12]. A similar conclusion can be drawn for the SDR metric from Figure 9. However, the NMF approach seems not good at improving the segmental SNR, even for the transient noise types as shown in

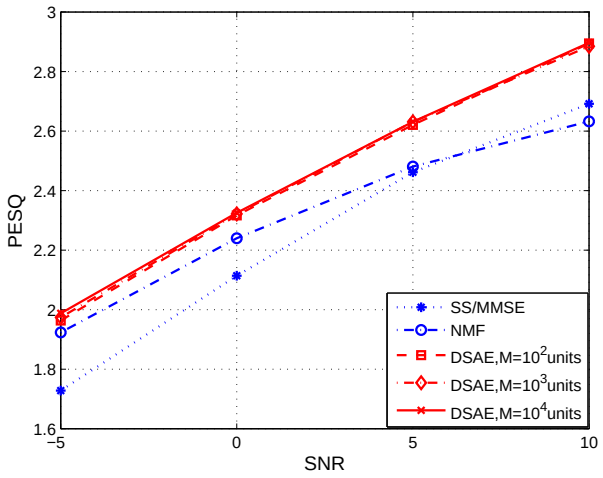

Fig. 5. The PESQ scores of SS/MMSE, NMF and SDAE at four different input SNR levels. For each condition, the numbers are the mean values over all the 20 noise types.

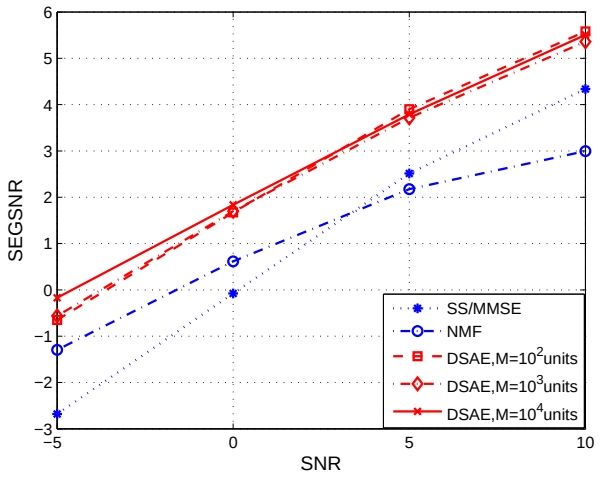

Fig. 7. The segmental SNR (SEGSNR) values of SS/MMSE, NMF and SDAE at four different input SNR levels. For each condition, the numbers are the mean values over all the 20 noise types.

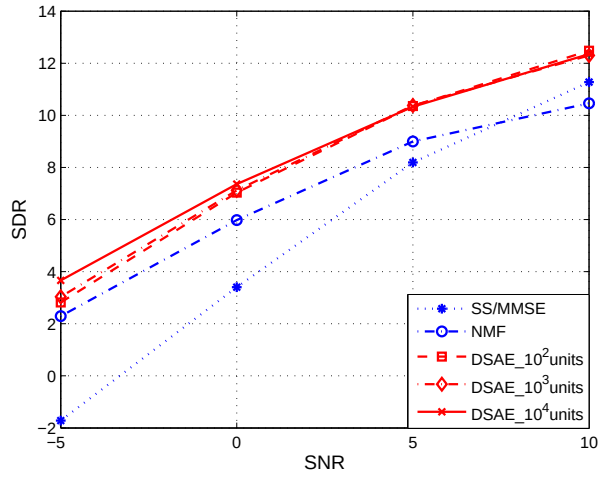

Fig. 6. The SDR values of SS/MMSE, NMF and SDAE at four different input SNR levels. For each condition, the numbers are the mean values over all the 20 noise types.

Figure 10. In [12], the *bird* noise was classified as a kind of transient noise, but in our experiments NMF did not show its expected superiority on this noise type. This may be due to the frequency fluctuations in the bird sounds which are difficult to be described by a couple of bases.

For relatively stationary noise types like *factory2* and *volvo*, SS/MMSE also performed good by yielding high PESQ scores. By considering all the three metrics, we conclude that significant improvements are observed by using SDAE over the SS/MMSE and NMF baselines.

G. The Method's Ability on Modeling Non-Stationary Noises

Table I presents the improvements on PESQ and SDR of SDAE and the super-Gaussian HMM [17] with respect to the noisy speech. The results were mean values over three noise types (*babble*, *factory1* and *freeway*). SDAE outperformed super-Gaussian HMM on the SDR metric for all the noise levels. For the PESQ metric, SDAE showed superiority over super-Gaussian HMM at low SNR levels of -5 dB and 0 dB.

TABLE I
COMPARISON OF THE PROPOSED METHODS AND THE SUPER-GAUSSIAN HMM IN [17]. THE RESULTS ARE THE AVERAGES OVER THE THREE NOISE TYPES *babble*, *factory1* AND *freeway*.

input SNR (dB)		-5	0	5	10
ΔSDR	Super-Gaussian HMM	5.8	5.6	4.8	**3.9**
	NMF	6.8	6.0	3.7	2.1
	SDAE	**8.1**	**7.2**	**5.5**	2.9
ΔPESQ	Super-Gaussian HMM	0.20	0.33	**0.38**	**0.41**
	NMF	0.23	0.35	0.28	0.21
	SDAE	**0.33**	**0.39**	**0.38**	0.34

Table II presents the improvements on PESQ of SDAE and the adaptive HMM [16] with respect to the noisy speech. The results are mean values over two noise types (*car+phone* and *formula1*). SDAE outperforms adaptive HMM on the PESQ metric for all the noise levels.

TABLE II
COMPARISON OF THE PROPOSED METHODS AND THE ADAPTIVE HMM IN [16]. THE RESULTS ARE THE AVERAGE IMPROVEMENTS ON PESQ WITH RESPECT TO SPECTRAL SUBTRACTION OVER THE TWO NOISE TYPES *Car+Phone* AND *formula1*.

input SNR (dB)	-5	0	5	10
Adaptive HMM	0.34	0.35	0.33	0.20
NMF	0.37	0.26	0.27	0.10
SDAE	**0.42**	**0.37**	**0.36**	**0.34**

In Table III, we compared our method with a recently proposed DNN approach in [30]. The DNN approach was trained in a supervised way to learn a mapping function from noisy to clean speech signals. A large noise collection containing 104 noise types was utilized to synthesize training data of noisy speech. Given many possible combinations of speech and noise types for training, the learned DNNs were

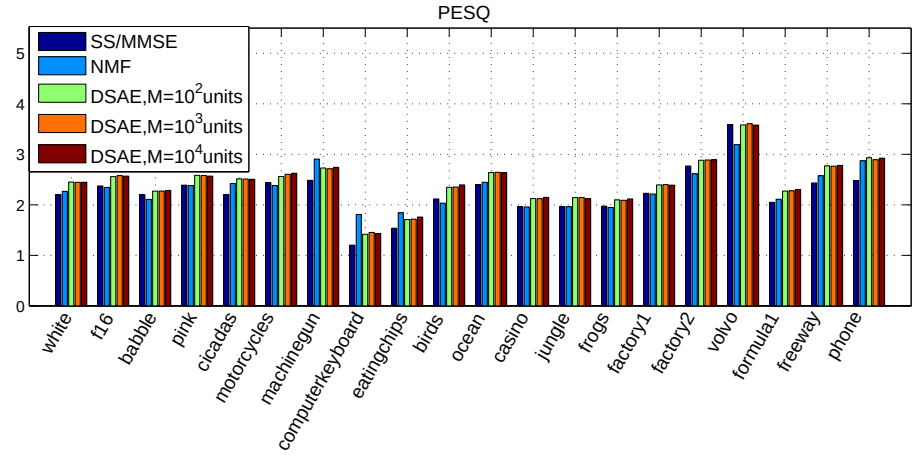

Fig. 8. The PESQ scores of SS/MMSE, NMF and SDAE for the 20 noise types. For each noise type, the numbers are the mean values over four input SNR conditions, i.e. from -5 dB to 10 dB spaced by 5dB.

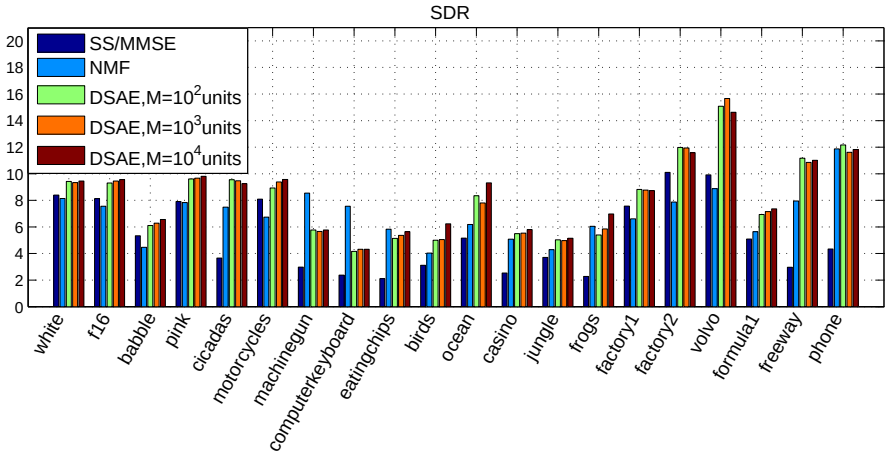

Fig. 9. The SDR values of SS/MMSE, NMF and SDAE for the 20 noise types. For each noise type, the numbers are the mean values over four input SNR conditions, i.e. from -5 dB to 10 dB spaced by 5dB.

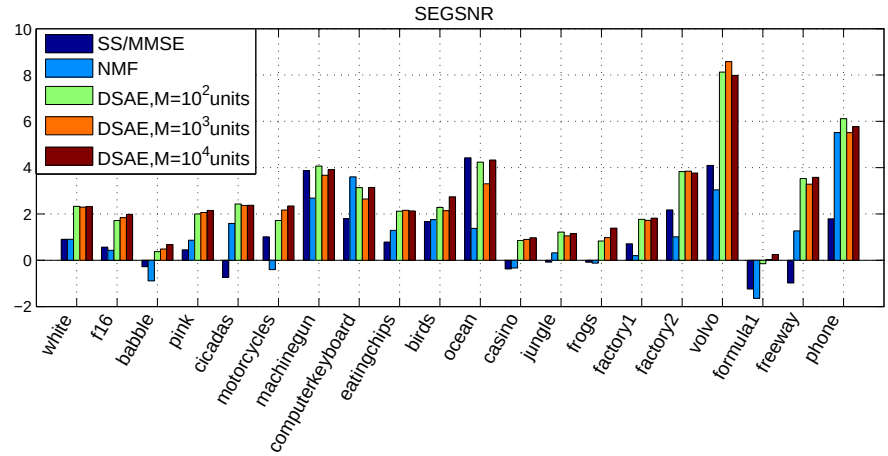

Fig. 10. The segmental SNR (SEGSNR) values of SS/MMSE, NMF and SDAE for the 20 noise types. For each noise type, the numbers are the mean values over four input SNR conditions, i.e. from -5 dB to 10 dB spaced by 5dB.

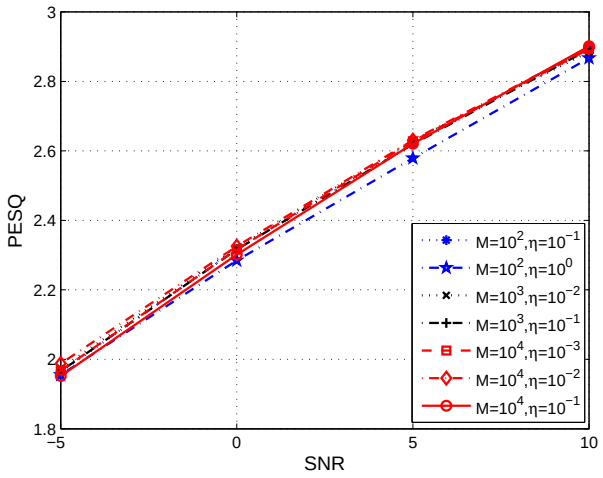

Fig. 11. The PESQ scores of the proposed SDAE with various parameter settings. For each case, the numbers are the mean values over all the 20 noise types.

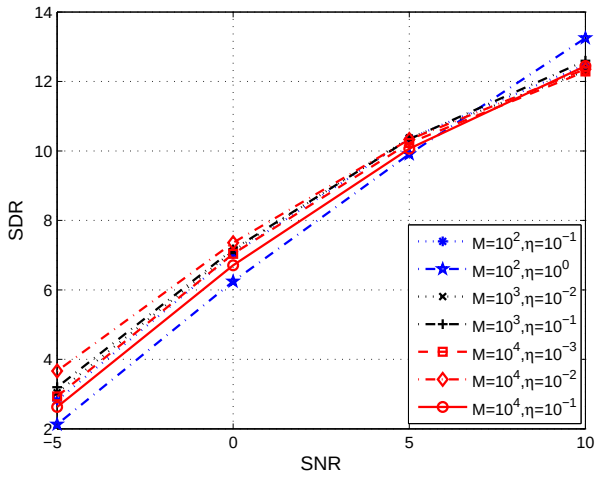

Fig. 12. The SDR values of the proposed SDAE with various parameter settings. For each case, the numbers are the mean values over all the 20 noise types.

expected to handle unseen noises in the enhancement stage. From Table III, we observed that our method worked slightly better than the DNN approach at -5dB and 0dB, but performed a bit worse than the DNN approach at 10 dB. However, our approach did not rely on any noise data collected beforehand, so its generalization ability to any unseen noise types could be stronger.

TABLE III
COMPARISON OF THE PROPOSED METHODS AND THE DEEP NEURAL NETWORK IN [30]. THE RESULTS ARE THE PESQ SCORES ON THE THREE NOISE TYPES *exhibition*, *destroyerengine*, AND *hfchannel* REPORTED IN [30].

	Exhibition		Destroyer engine		HF channel	
	DNN	SDAE	DNN	SDAE	DNN	SDAE
SNR10	**3.00**	2.84	**3.24**	3.10	**2.82**	2.78
SNR 5	**2.63**	2.58	**2.91**	2.88	2.52	**2.63**
SNR 0	2.24	**2.25**	2.55	**2.63**	2.24	**2.32**
SNR-5	1.80	**1.88**	2.17	**2.26**	1.92	**1.97**

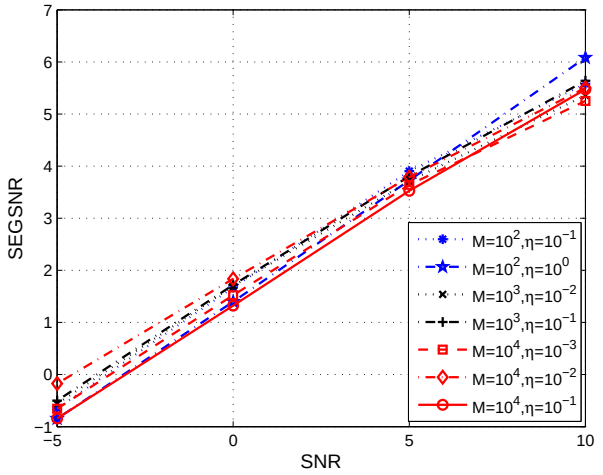

Fig. 13. The segmental SNR (SEGSNR) values of the proposed SDAE with various parameter settings. For each case, the numbers are the mean values over all the 20 noise types.

H. The Method's Sensitivity on the Configuration Parameters of the Noise DAE

The PESQ scores, SDR values and SEGSNR values of the SDAE with various parameter settings are given in Figure 11, Figure 12 and Figure 13, respectively.

The number of units in SDAE_NMF performs a role only at low SNR levels. $M = 10^4$ units with $\eta = 1\%$ units retained gave the best performance at the -5 dB condition. At low SNR levels, the total volume of the noise is high, a large amount of hidden units are required to cope with this. With the temporal variation of the noise, different frames hold different spectral properties which would thus reflected by activating a couple of different hidden units.

To evaluate the algorithm's sensitivity regarding α, we conducted experiments on SDAE with $M = 100$ units and activating top $\eta = 10\%$ units for each fames. The results on PESQ scores, SDR values and SEGSNR values are given in Figure 14, Figure 15 and Figure 16, respectively. From the figures, we see that $\alpha = 0.1$ is a good choice. A larger one, say $\alpha = 1$, will be hazard; while smaller ones (e.g. $\alpha = 0.01$ or $\alpha = 0$) will not improve the baselines so much.

V. CONCLUSION

Separable deep auto encoder (DAE) was proposed to estimate unseen noise spectrum for speech enhancement. A DAE was first trained for clean speech spectrum reconstruction. An additional DAE was introduced to model the unseen noise spectrum with the constraint that the sum of the outputs of the two DAEs is equal to the input noisy speech spectrum. To help the inverse problem yield meaningful solutions, nonnegative matrix factorization (NMF) was imposed to constrain the speech spectrum reconstruction. New multiplicative algorithms were investigated to optimize the problem of DAE with

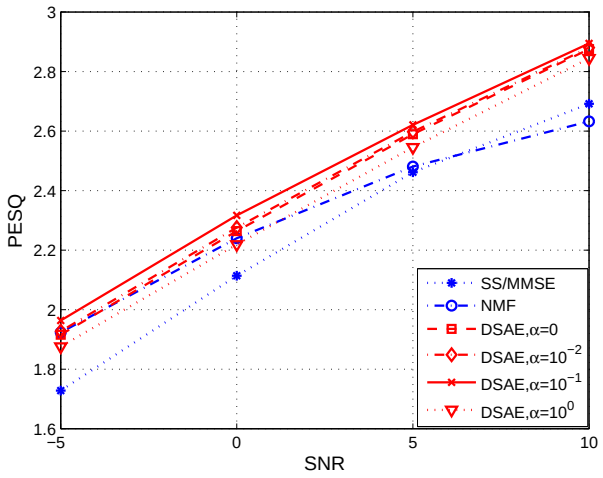

Fig. 14. The PESQ scores of the proposed SDAE with the regularization parameter α. The SDAE contains $M = 100$ units and takes top $\eta = 10\%$ units for each fames. For each case, the numbers are the mean values over all the 20 noise types.

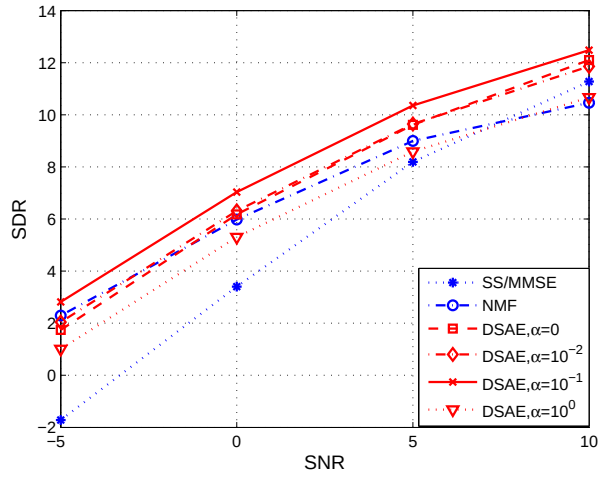

Fig. 15. The SDR values of the proposed SDAE with the regularization parameter α. The SDAE contains $M = 100$ units and takes top $\eta = 10\%$ units for each fames. For each case, the numbers are the mean values over all the 20 noise types.

NMF constraints. Experimental evaluation on PESQ, SDR and segmental SNR on the TIMIT dataset contaminated by 20 types of unseen noises demonstrated the superiority of the proposed approach over the traditional baselines including SS/MMSE, NMF and HMM. Due to the heavy computation budget, it is currently difficult to apply the proposed method for real-time implementations. However, for post-processing of recorded audio files, our methods can have advantages given the experimental results presented in the paper.

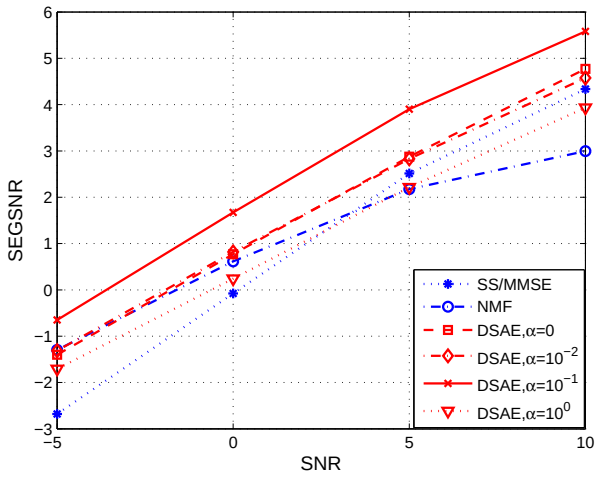

Fig. 16. The segmental SNR (SEGSNR) values of the proposed SDAE with the regularization parameter α. The SDAE contains $M = 100$ units and takes top $\eta = 10\%$ units for each fames. For each case, the numbers are the mean values over all the 20 noise types.

REFERENCES

[1] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhanceing speech corrupted by colored noise," in *ICASSP*, 2002.

[2] K. Paliwal, K. Wjcicki, and B. Schwerin, "Single-channel speech enhancement using spectral subtraction in the short-time modulation domain," *Speech Communication*, vol. 52, no. 5, pp. 450–475, 2010.

[3] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust. Speech Signal Process*, vol. 33, pp. 443–445, 1985.

[4] P. C. Loizou and S. Member, "Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum," in *IEEE Transactions on Speech Audio Processing*, 2005, pp. 857–869.

[5] T. Gerkmann and R. C. Hendriks, "Unbiased mmse-based noise power estimation with low complexity and low tracking delay," *IEEE Trans Audio, Speech and Language Processing*, vol. 20, pp. 1383–1393, 2012.

[6] P. K. Ghosh, A. Tsiartas, and S. S. Narayanan, "Robust voice activity detection using long-term signal variability," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 3, pp. 600–613, 2011.

[7] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan, "Hmm-based strategies for enhancement of speech signals embedded in nonstationary noise," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 6, no. 5, pp. 445–455, 1998.

[8] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 163–176, 2006.

[9] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models." *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[10] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in

INTERSPEECH, 2013, pp. 436–440.

[11] P. Smaragdis, "Non-negative matrix factor deconvolution: extraction of multiple sound sources from monophonic inputs," in *In Fifth International Conference on Independent Component Analysis*, Sep. 2004, pp. 494–499.

[12] Z. Chen and D. P. Ellis, "Speech enhancement by sparse, low-rank, and dictionary spectrogram decomposition," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[13] C. Fvotte, J. L. Roux, and J. R. Hershey, "Non-negative dynamical system with application to speech and audio," in *ICASSP*, 2013, pp. 3158–3162.

[14] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2140–2151, 2014.

[15] D. Y. Zhao, W. B. Kleijn, A. Ypma, and B. de Vries, "Online noise estimation using stochastic-gain hmm for speech enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 4, pp. 835–846, 2008.

[16] J. Bai and M. Brookes, "Adaptive hidden markov models for noise modelling," in *In 19th European Signal Processing Conference (EUSIPCO 2011)*, August 2011, pp. 494–499.

[17] N. Mohammadiha, R. Martin, and A. Leijon, "Spectral domain speech enhancement using hmm state-dependent super-gaussian priors," *IEEE Signal Processing Letters*, 2013.

[18] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, January 2015.

[19] J. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv*, August 2014.

[20] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *ICASSP*, 2013.

[21] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[22] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *ICASSP*, 2014.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, , and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arxiv*, 2014. [Online]. Available: http://arxiv.org/abs/1207.0580

[24] http://www.zedge.net/ringtones/.

[25] http://www.soundsnap.com/.

[26] http://www.freesound.org/.

[27] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq) - a new method for speech quality assessment of telephone networks and codes," in *ICASSP*, 2001, pp. 749–752.

[28] E. Vincent, C. Fevotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio Speech Lang. Process*, vol. 14, pp. 1462–1469, 2006.

[29] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox, 2012.

[30] "An experimental study on speech enhancement based on deep neural networks," *IEEE SIGNAL PROCESSING LETTERS*, vol. 21, no. 1, January 2014.

Meng Sun received his Ph.D. degree from the Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven in November, 2012. He is now a researcher at the Lab of Intelligent Information Processing of PLA University of Science and Technology, China. His research interests are speech processing, unsupervised/semi-supervised machine learning and sequential pattern recognition.

Xiongwei Zhang received his Ph.D. degree from the PLA University of Science and Technology, China. He is now a professor in the Lab of Intelligient Information Processing in the same university. His rearch interests are speech coding, speech enhancement and image processing.

Hugo Van hamme (M'92) received the degree of electrical engineer from the Vrije Universiteit Brussels, Brussels, Belgium, in 1987, the M.S. degree from Imperial College, London, U.K., in 1988, and Ph.D. degree from Vrije Universiteit Brussels in 1992. In 1993, he joined Lernout and Hauspie as a Senior Researcher. Later, he headed the speech recognition research activities in Belgium at this company. In 2001, he joined ScanSoft as a Manager of Research and Engineering for the Automotive Division. Since 2002, he is affiliated full-time as a Professor at Katholieke Universiteit Leuven, Leuven, Belgium. His main research interests are robust speech recognition, computational models of language acquisition, and computer-assisted learning.

Thomas Fang Zheng (M'99-SM'06) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 1997. He is now a Research Professor and Director of the Center for Speech and Language Technologies, Tsinghua University. His research focuses on speech and language processing. He has published more than 230 papers. Dr. Zheng plays active roles in a number of communities, including the Chinese Corpus Consortium (council chair), the Standing Committee of China's National Conference on Man-Machine Speech Communication (chair), Subcommittee 2 on Human Biometrics Application of Technical Committee 100 on Security Protection Alarm Systems of Standardization Administration of China (deputy director), the Asia-Pacific Signal and Information Processing Association (APSIPA) (Vice-President and Distinguished Lecturer 2012- 2013), Chinese Information Processing Society of China (council member and Speech Information Subcommittee Chair), the Acoustical Society of China (council member), and the Phonetic Association of China (council member). He is an Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing and the APSIPA Transactions on Signal and Information Processing. He is on the editorial board of Speech Communication, Journal of Signal and Information Processing, Springer Briefs in Signal Processing, and the Journal of Chinese Information Processing.