Modulation Filter Learning Using Deep Variational Networks for Robust Speech Recognition

Purvi Agrawal, *Student Member, IEEE,* and Sriram Ganapathy, *Senior Member, IEEE*

***Abstract*—The performance of a typical speech recognition system is degraded in the presence of extrinsic sources like noise and due to the recording artifacts like reverberation. The principle of modulation filtering attempts to remove the spectro-temporal modulations of the speech signal that are more susceptible to noise while preserving the key modulations for speech recognition. While traditional approaches use modulation filters that are hand-crafted, we propose a novel method for modulation filter learning using deep variational models in this paper. Specifically, we pose the filter learning problem in a deep unsupervised generative modeling framework where the convolutional filters in the variational autoencoder capture the important speech modulations. The two dimensional (2-D) modulation filters, learned using the deep variational networks in the joint-spectro temporal domain, are used to process the spectrogram features for speech recognition task. Several speech recognition experiments are performed on a set of tasks consisting of additive noise with channel artifacts (Aurora-4), reverberation (REVERB Challenge) and additive noise with reverberation (CHiME-3). In these experiments, the proposed modulation filter learning framework shows significant improvements over the baseline features as well as various other noise robust front-ends (average relative improvements of 7.5% and 20% over the baseline features on the Aurora-4 and CHiME-3 databases respectively). Furthermore, the proposed method is also shown to be of considerable benefit for semi-supervised ASR applications. For example, on Aurora-4 database we observe an average relative improvement of 25% over the baseline system using 30% labeled training data.**

***Index Terms*—Unsupervised filter learning, Deep Variational Autoencoder, Modulation filtering, Noise robust speech recognition.**

I. Introduction

EVEN with several advancements in acoustic modeling for automatic speech recognition (ASR) using deep learning [1] and sequence modeling [2], the performance of the ASR systems are highly degraded in the presence of extrinsic sources like noise and telephone channel distortions. Further, the far-field speech recording conditions also distort the signal with reverberant artifacts which transpires as a smearing of the temporal envelopes of the speech signal. In particular, the degradation due to reverberation is a notable challenge in the development of a real world application of hands free ASR [3]. For example, Peddinti *et al.,* [4] reports a 75% rel. degradation in word error rate (WER) when far-field array microphone signals are used instead of the headset microphones in the ASR systems, both during training and testing.

The performance degradation in noisy and reverberant conditions can be partly addressed by training on multi-conditioned data (consisting of noisy training data from multiple real/simulated environments) [5]. Further, enhancement methods like mask estimation [6], spectral subtraction [7], power normalization [8], Hilbert envelope compensation [9] and dereverberation methods like weighted prediction error [10] have shown benefits in improving the performance of ASR. However, it has been observed that the performance of an ASR system in noisy environments is considerably worse compared to clean training/test conditions even with multi-condition training and feature compensation methods. In this paper, the issue of robustness in feature representation is addressed using an unsupervised feature learning method.

The principle of modulation filtering in robust automatic speech recognition (ASR) is based on enhancing perceptually relevant regions of the modulation spectrum (the two dimensional Fourier transform of a patch of the speech spectrogram captures the spectro-temporal modulation content) while suppressing the regions susceptible to noise. This is partly inspired by human perceptual studies relating to the importance of temporal modulations (called rate; rate frequency measured in Hertz) and spectral modulations (called scale; scale frequency measured in cycles/octave) [11]. The evidence of spectro-temporal modulations in the perception of complex sounds was shown with experiments in which systematic degradation of the speech signal were correlated with the gradual loss of intelligibility [12]. It has also been shown that important information for speech perception lies in the $1-16$ Hz range of the rate frequencies [13]. For ASR applications, one of the earliest use of temporal modulations was the RASTA filtering approach [14]. The spectro-temporal modulation filtering has also been explored for voice activity detection [15] and for phoneme recognition in noise [16]. An unsupervised learning of sparse spectro-temporal coding of sounds has been attempted in [17], [18]. In addition, learning of natural sounds/codes in the auditory cortex has also been carried out in unsupervised manner in [19], [20]. A supervised data-driven approach using the linear discriminant analysis (LDA) has also been attempted for deriving temporal modulation filters [21].

In this paper, we propose a new approach to learn spectral and temporal modulation filters purely from a variational generative modeling perspective. In particular, we develop a filter learning method using the speech spectrogram in conjunction with a two-dimensional (2-D) convolutional vari-

P. Agrawal and S. Ganapathy are with the Learning and Extraction of Acoustic Patterns (LEAP) lab, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India, 560012. e-mail: {purvia,sriramg}@iisc.ac.in.

This work was funded by grants from the Department of Science and Technology (DST) Early Career Award and the Pratiksha Trust Young Investigator Award.

Manuscript received October 08, 2018 and revised on April, 25, 2019.

ational autoencoder (CVAE) [22]. The encoder learns the distribution of the latent representation and the decoder attempts to reconstruct the original data back from a sample of the latent representation generated from the encoder distribution. The filters learned from the input speech spectrogram in the initial convolutional layer of CVAE can provide important cues regarding the useful spectro-temporal modulations of speech. In this paper, the modulation filters learned from CVAE are applied on the input spectrogram to derive features for speech recognition. Previous attempts to filter learning for speech include a data-driven supervised approach for mel filter bank learning from raw speech waveforms [23], [24] and unsupervised approach in [25], [26]. A 1-D and 2-D modulation filter learning using residual approach was also previously attempted using the restricted Boltzmann machine (RBM) architecture [27], [28].

In this work, the 2-D spectro-temporal modulation filters learned from the CVAE model in an unsupervised fashion are used to process the speech spectrogram for deriving robust spectrogram representations. The processed spectrogram representations are used in deep neural network based automatic speech recognition systems. We perform ASR experiments on several tasks involving additive/channel noise (Aurora-4), reverberation effects (REVERB Challenge) [29] and additive noise with reverberation (CHiME-3) [30]. In these experiments, the proposed filter learning approach provides significant improvements in terms of WER over the baseline mel filter bank features and various other noise robust front-ends proposed in the past.

The rest of the paper is organized as follows. In Sec. II, we describe the theory of variational modeling in autoencoder networks. The use of convolutional variational autoencoder (CVAE) for filter learning from speech signal is discussed in Sec. III. Sec. IV describes various ASR experiments and results. This is followed by a discussion of the proposed approach in Sec. V. We also analyze the importance of various model parameters and report the performance of semi-supervised speech recognition experiments in Sec. V. In Sec. VI, we provide a brief summary of the paper.

II. Variational Autoencoder (VAE)

A VAE is a modification of an AE that consists of an encoder and a decoder [22]. In a traditional AE, the encoder estimates latent variables (bottleneck representations) [31] and the decoder, also based on deep networks, then reconstructs the observation variables from the latent variables. The VAE model assumes that the samples of latent representation can be drawn from a standard normal distribution, i.e $\mathcal{N}(0; I)$ [22]. The encoder estimates the parameters of the latent data distribution that approximates the posterior distribution of the latent vector given the data. The decoder then samples from the approximate distribution and attempts to reconstruct the original data back. The variational autoencoder networks have shown promising results in tasks like image captioning [32], text generation [33] and voice conversion [34].

Let x denote an observation vector, and z denote the latent vector. Let denote the set of parameters for the decoder network. The aim of the VAE network is to maximize the probability of each x in the training set under the generative process, according to

$$p\ (x) = \int p\ (x|z)p(z)dz: \tag{1}$$

The model involves a two step process: (1) a value z is generated from prior distribution $p(z)$; (2) a value x is generated from conditional distribution $p\ (x|z)$. However, in the assumed generative model with a decoder neural network, the function $p\ (x)$ is not always differentiable w.r.t. due to the intractable integral in Eq. 1; therefore cannot be optimized directly. In addition, the posterior distribution of the latent vector $p\ (z|x) = p\ (x;z)$=$p\ (x)$ can also be intractable, thereby making the application of Expectation Maximization (EM) algorithm difficult.

The VAE framework resolves these problems based on a variational lower bound method [22]. A new function $q\ (z|x)$ (probabilistic encoder with encoder parameters) is introduced that can take value of x and give a distribution over z values. In other words, the function $q\ (z|x)$ approximates the true posterior distribution $p\ (z|x)$.

The key idea behind the variational autoencoder is to attempt to sample values of z that are likely to have generated x, and lower bound the value of $p\ (x)$ using those. For achieving this, the encoder and decoder parameters, and , respectively, are trained by maximizing the lower bound $\mathcal{L}(\ ;\ ;x)$ of the marginal likelihood $\log p\ (x)$, given as,

$$\mathcal{L}(\ ;\ ;x) = -D_{KL}[q\ (z|x)||p(z)] + \mathbb{E}_{z|x\ \ q_\phi}[\log p\ (x|z)] \tag{2}$$

Given the parameters of the encoder network, (x) and $^2(x)$ which are the mean and variance parameters of $q\ (z|x)$ - we can sample from $\mathcal{N}(\ \ (x); diag(\ ^2\ (x)))$ by first sampling $\sim \mathcal{N}(0; I)$, then computing $z = \ \ (x) + diag(\ \ (x))$, shown schematically in Fig. 1.

A. VAE training procedure

With the encoder and decoder as deep neural networks:

An observation vector x is given as input to an encoder network that estimates $q\ (z|x)$. The encoder outputs a mean vector $\ \ (x)$ and a variance vector $^2(x)$ which are used to sample a latent vector z.

The latent vector z is sampled according to the Gaussian distribution with $\ \ (x)$ and $^2(x)$ using the reparameterization trick (sample according to standard normal distribution and then use the encoder mean and variance parameters to transform the variable to match the distribution $q\ (z|x)$).

The decoder $p\ (x|z)$ generates x from z assuming Gaussian distribution.

Stochastic gradient ascent is performed on the lower bound $\mathcal{L}(\ ;\ ;x)$ w.r.t. model parameters (,). The negative of the first term in Eq. 2, $D_{KL}[q\ (z|x)||p(z)]$ is termed as 'latent loss' (E_{Latent}) which is the KL divergence between two multivariate Gaussian distributions. The second term in Eq. 2, $\mathbb{E}_{z|x\ \ q_\phi}[\log p\ (x|z)]$, with Gaussian assumptions on $p\ (x|z)$, reduces to the negative

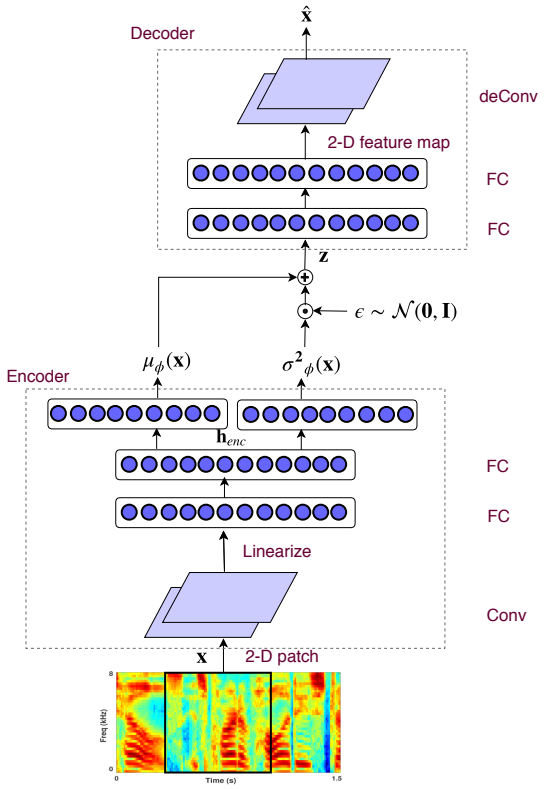

Fig. 1. Block schematic of filter learning with CVAE. Here FC denotes fully connected layer and Conv, deConv denotes convolution and deconvolution layer, respectively.

of mean square error (MSE) loss (E_{MSE}), typically used in conventional autoencoder. Thus, VAE loss function can also be viewed as a minimization of regularized MSE loss where the regularization comes from the KL divergence term. In our implementation of VAE, we also weigh the two losses (MSE and Latent) differently to control the regularization term. The parameters of the VAE model are updated using stochastic gradient descent (SGD).

B. Comparing VAE with other deep generative models.

The VAE is a generative model which minimizes the MSE of the reconstructed data along with a latent loss. The two other popular models for generative modeling in neural network framework are the restricted Boltzmann machine (RBM) [35] and the generative adversarial networks (GAN) [36]. The RBM model [35] also uses a latent representation similar to the VAE. The model assumes a Boltzmann distribution for the joint density function of the observation and latent variable. For the model parameter learning, a maximum likelihood approach is used where a Gibbs sampling framework is employed. The GAN models [36] also aim to use a latent data distribution to generate the observed data. The model uses a discriminative loss function (fake versus real) to further correct the generative model. The conventional GAN uses an independent distribution to generate the latent vector and does not use the observation data to generate the latent vector (unlike the VAE model).

TABLE I
THE ARCHITECTURE OF THE CVAE MODEL USED FOR FILTER LEARNING.

Number of layers - encoder	Conv: 1, FC: 2
Number of layers - decoder	FC: 2, deConv: 1
Number of kernels in Conv/deConv	2 (kernel size: 5×5)
Number of nodes in FC	6000
Activation function	tanh
Latent Vector $\boldsymbol{z}$ Dimension	5000
Mini-batch size	1200
Optimization	Adam [37]
Learning rate	0.0001

III. CONVOLUTIONAL VAE AND FILTER LEARNING

The block diagram of the VAE model used for filter learning is shown in Figure 1. The convolutional VAE (CVAE) used in this work replaces the fully connected layer(s) in the encoder and decoder networks with convolution layer(s). The use of CVAE is motivated by the goal of learning modulation filters in an unsupervised manner. The kernels (convolutional filters) of the deep CVAE trained using spectrogram input are interpreted as the modulation filters learned from the data that characterize the key modulations required to generate speech. We train the CVAE in multi-condition fashion with a small number of filters (we use two filters in the convolutional layer). This way, the model is constrained to primarily learn the speech distribution while ignoring the noise distribution.

A. Implementation of CVAE for filter learning

As outlined in Figure 1, the input to CVAE are the 2-D patches of log mel spectrograms. The mel spectrogram is computed using short-time Fourier transform of speech signal with 25 ms frame length and shift of 10 ms, and warping the frequency axis with 40 mel-bands. The dimension of the 2-D patch of the spectrogram at the input of CVAE is 150×40 (equivalent to 1.5s of speech from 40 mel bands). Table I gives the details of the CVAE architecture used in this work. The first layer of the 'encoder' is a convolutional layer with number of kernels $= 2$. The size of the kernels is 5×5 and is constrained to be of rank$= 1$ in order to learn separable spectro-temporal filters [38]. Hence, the filters W_1 and W_2 of the convolutional layer can be decomposed as the outer product of temporal modulation 'rate' filter r and spectral modulation 'scale' filter s as $W_1 = (r_1 s_1{}^T)$ and $W_2 = (r_2 s_2{}^T)$, respectively.

The output of convolutional (Conv) layer is linearized and fed to fully-connected (FC) layers of the encoder. The latent vector $\boldsymbol{z}$ is calculated from the encoder output as discussed in Section II. The decoder then reconstructs the 2-D patch from the latent vector $\boldsymbol{z}$ by reversing the steps in the encoder (fully connected layers followed by a deconvolution layer [31]).

In our implementation of the CVAE, we modify the loss function as follows,

$$E_{Total} = E_{MSE} + E_{Latent} + (E_{fr} + E_{fs}) + E_{enc} \quad (3)$$

with,

$$E_{MSE}(x; \hat{x}) = ||x - \hat{x}||_2^2 \;\; ; \;\; E_{enc}(h_{enc}) = ||h_{enc}||_1 \quad (4)$$

$$E_{Latent} = D_{KL}(q\ (z|x); p(z)) \quad (5)$$

$$E_{fr}(r_1; r_2) = ||r_1 * r_2||_2^2 \;\; ; \;\; E_{fs}(s_1; s_2) = ||s_1 * s_2||_2^2 \quad (6)$$

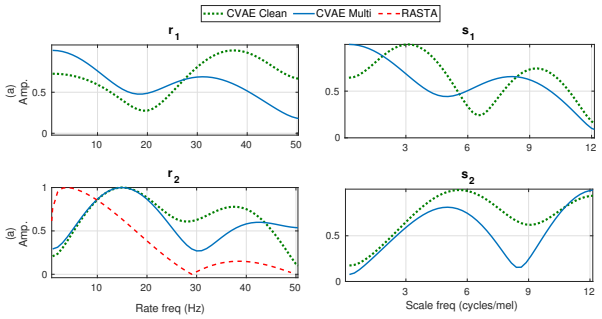

Fig. 2. Two sets of rate ($r_1; r_2$) and scale filters ($s_1; s_2$) learned from the CVAE model using the clean condition and multi-condition Aurora-4 dataset. The rate filters have low-pass and band-pass characteristics in this case. The RASTA filter is also shown in the r_2 plot for reference.

where $*$ denotes convolution operation. The L2-norm of convolution of filters are introduced primarily to avoid learning redundant filters (filters with highly overlapping frequency responses). Note that minimizing the convolution loss of filters r_1 and r_2 (or s_1 and s_2) is equivalent to minimizing the product of frequency response of these filters. We found that L2 convolution norm loss constraint helps in generating modulation filters that cover the broad modulation range of speech signal. The L1 norm loss (E_{enc}) encourages sparse representation in hidden latent dimensions. This is motivated by the success of sparse autoencoder in speech applications [39]. The sparse regularization term is also beneficial in this case as the latent dimensions in CVAE are quite high (5000). The scaling factors ; ; ; are hyper parameters which are set based on validation experiments. Note that the original formulation of VAE uses (; = 1) and (; = 0). In this paper, we use a modified VAE cost function in addition to the sparsity loss and convolution loss[1]. The benefits of this modified loss are highlighted in Sec. V.

B. Filter Responses

The filters $r_1; s_1; r_2; s_2$ are iteratively updated using the gradients of the total loss function in Eq. 3. The CVAE is trained using multi condition and clean data of different databases separately. We start the network training with random initialization of the filters and the weights and allow the CVAE to learn modulation filter characteristics from data. Fig. 2 shows the normalized magnitude frequency response of the filters learned using multi-condition and clean Aurora-4 database (details of the Aurora-4 dataset are given in Sec. IV) respectively. Since each 2-D filter is constrained to be rank-1, the frequency response of individual rate and scale components of the filters can be separately plotted, shown in Figure 2. The x axes for the rate and scale filters are rate frequencies (measured in Hz) and scale frequencies (measured in cycles/mel) respectively. The value of scaling factors in cost function of CVAE used in this case are = 1:0, = 0:5, = 0:5 and = 0:1.

In our analysis, we find that the two rate filters learned from the input mel spectrogram have invariably low-pass and band-pass characteristics in multi-condition data, while it is band-stop and band-pass for clean data (Fig. 2). The scale filters jointly span the entire spectral modulation space. We assume that the filters will learn the common underlying representation of all types of input noisy speech, which would be dominated by clean speech. The second row of Figure 2 also shows the comparison with the RASTA filter [14]. As seen here, the learnt data-driven rate filter somewhat resembles the perceptual knowledge driven RASTA filter. Also, the range of modulations captured by $r_1; r_2$ and $s_1; s_2$ are quite similar to the modulation filters found in human perceptual studies [11]. This is interesting in the sense that, even with random initialization, the data-driven generative modeling of a corpus of speech using the framework in CVAE can yield filters that are broadly similar to filters found in various perceptual studies on modulations. The frequency response of filters learned from other datasets is discussed in Sec. V.

[1]The implementation of the proposed filter learning approach is available at https://github.com/iiscleap/CVAE_FilterLearning

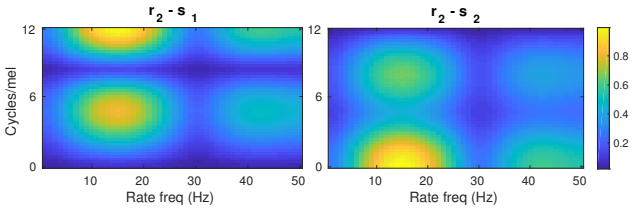

Fig. 3. The two 2-D filters ($r_2 - s_1$) and ($r_2 - s_2$) used in feature extraction for ASR in Aurora-4 multi-condition database.

C. Feature extraction for ASR

The features for ASR are derived by filtering the log mel spectrogram using filters learned from the proposed approach. In this work, we select the rate filter with bandpass characteristic as it has been observed earlier to be crucial for ASR performance [14], [27], while all the scale filters spanning spectral modulation space are used for ASR. Detailed analysis on filter selection and number of filters is given in Sec. V. The 2-D filter responses for the filters used in multi-condition ASR for the Aurora-4 dataset are shown in Figure 3. The log mel spectrograms are filtered using filters ($r_2 - s_1$) and ($r_2 - s_2$) separately and are concatenated to derive features for ASR. This is motivated from the works in the past about neurophysiological evidence suggesting that processing of speech signals in the brain happens along parallel pathways and encode complementary information in the signal [40], [41]. The proposed features (proposed as well as all the other baseline features in comparison) are mean-variance normalized at the utterance level before the acoustic model training. In all the ASR experiments, we do not perform any speaker level normalization.

IV. Experiments and Results

A. Database

*1) **Aurora-4**:* The Wall Street Journal (WSJ) Aurora-4 corpus consists of continuous read speech recordings of 5000 words corpus, recorded under clean and noisy conditions (street, train, car, babble, restaurant, and airport) at $10-20$ dB

SNR. The training data has 7138 clean and multi condition recordings from 84 speakers separately. The validation data has 1206 clean and multi condition recordings respectively. The test data has 330 recordings (8 speakers) for each of the 14 test conditions (clean and noisy). The test data is classified into four groups, A - clean data, B - noisy data, C - clean data with channel distortion, and D - noisy data with channel distortion.

*2) **REVERB Challenge**:* The REVERB challenge [29] dataset consists of multi-channel reverberant data of continuous reverberant speech from WSJCAM0 corpus. This database consists of 7861 recordings from 92 training speakers, 1488 recordings from 20 development test (dev) speakers and 2178 recordings from two sets of 14 evaluation test (eval) speakers, with each speaker providing about 90 utterances. These recordings were carried out with two sets of microphone - head mounted as well as desk microphone positioned about half a meter from the speaker's head.

The database used in this work consists of three subsets: training data set (Train) for multi condition training using simulated reverb data (multi-channel beamformed data from 8 microphones using the BeamFormIt tool [42]), a simulated test dataset (Sim) and a naturally reverberant recording of the test dataset (Real).

*3) **CHiME-3 Challenge**:* The CHiME-3 corpus for ASR contains multi-microphone tablet device recordings from everyday environments, released as a part of 3rd CHiME challenge [30]. Four varied environments are present, cafe (CAF), street junction (STR), public transport (BUS) and pedestrian area (PED). For each environment, two types of noisy speech data are present, real and simulated. The real data consists of 6-channel recordings of sentences from the WSJ0 corpus spoken in the environments listed above. The simulated data was constructed by artificially mixing clean utterances with environment noises. The training data has 1600 (real) noisy recordings - four speakers each reading 100 utterances in each of the four environments (i.e. $4 \times 4 \times 100$). These sentences were randomly selected from the 7138 utterance WSJ0 5k training data. The real data is supplemented by 7138 simulated utterances constructed by taking the full WSJ0 5k training set and mixing it with the separately recorded CHiME-3 noise backgrounds. We use the beamformed audio for ASR training and testing.

The development and test data consists of the same 410 and 330 utterances that make up the corresponding sets in the WSJ0 5k task. For each set, the sentences are read by four different talkers in the four CHiME-3 environments. In each environment, the set is split into four random partitions and each is assigned to a different talker. This results in 1640 (410×4) and 1320 (330×4) real development and test utterances in total. Identically-sized, simulated test sets are made by mixing recordings captured in the recording booth with the environmental noise recordings.

B. Kaldi ASR framework

The speech recognition Kaldi toolkit [43] is used for building the ASR. For the ASR experiments on Aurora-4, CHiME-3 and REVERB Challenge, we use a deep neural network (DNN) with 6 hidden layers and using splice of 10, i.e. 21 frames of input temporal context. The DNNs with sigmoid nonlinearity are pre-trained with RBM training (separate pre-trained models for each of the features). Then, the models are discriminatively trained using the training data with cross entropy loss. A hidden Markov model - Gaussian mixture model (HMM-GMM) system trained using MFCC features is used to generate the alignments for training the DNN based model. A tri-gram language model is used in the ASR decoding and the best language model weight is obtained from development set. This recipe follows the corpus release for the training and evaluation splits for all the datasets considered in this paper. The performance of the ASR system is analyzed using word-error-rate (WER). We compare the ASR performance of the proposed modulation filtering approach (CVAE) with traditional mel filter bank energy (MFB) features, power normalized filter bank energy (PFB) features [8], advanced ETSI front-end (ETS) [44], RASTA features (RAS) [14], LDA based features (LDA) [45], and MHEC features (MHE) [9]. In particular, the RASTA features (RAS) and LDA features are included as they both perform modulation filtering in the temporal domain using a knowledge driven filter and a supervised data-driven filter, respectively.

TABLE II
WORD ERROR RATE (%) IN AURORA-4 DATABASE FOR CLEAN TRAINING CONDITION WITH VARIOUS FEATURE EXTRACTION SCHEMES AND THE PROPOSED CVAE MODULATION FILTERING APPROACH.

Cond	MFB	PFB	ETS	RAS	LDA	MHE	CVAE
A. Clean with same Mic							
Clean	3.4	3.3	3.2	3.5	3.7	3.5	**3.0**
B: Noisy with same Mic							
Airport	21.9	18.3	15.0	19.3	23.2	19.5	17.0
Babble	19.6	16.0	15.5	19.9	21.0	17.7	15.9
Car	8.0	6.2	9.8	7.9	8.7	7.9	6.8
Rest.	24.9	22.9	20.5	23.0	27.0	23.2	21.2
Street	19.5	17.8	19.5	18.7	20.8	18.1	17.1
Train	19.8	16.3	17.4	19.4	20.1	17.9	17.6
Avg.	18.9	16.2	16.3	18.0	20.1	17.4	**15.9**
C: Clean with diff. Mic							
Clean	15.3	**11.7**	14.5	16.0	15.9	14.6	13.6
D: Noisy with diff. Mic							
Airport	40.1	36.4	31.4	39.2	40.4	38.7	35.5
Babble	37.3	34.2	32.1	38.5	36.8	36.8	34.1
Car	24.9	21.5	24.9	24.8	25.9	25.8	22.6
Rest.	39.6	39.0	35.4	39.1	41.0	39.3	36.4
Street	35.7	34.1	35.0	35.8	37.0	35.8	34.2
Train	35.6	31.8	33.2	36.4	36.7	35.9	35.3
Avg.	35.2	32.8	**32.0**	35.6	36.3	35.4	33.0
Avg. of all conditions							
Avg.	24.7	22.1	**21.9**	24.4	25.6	23.9	22.1

C. Results

The results of various ASR experiments on clean and multi-condition Aurora-4 dataset is shown in Table II and III respectively. The ASR results have also been separately reported for different noisy conditions (conditions A, B, C, D). As seen in Table II for clean training, the noise robust front-ends improve over the baseline mel-filter bank (MFB) performance. The proposed CVAE improves over the baseline

TABLE III
WORD ERROR RATE (%) IN AURORA-4 DATABASE FOR MULTI CONDITION TRAINING CONDITION WITH VARIOUS FEATURE EXTRACTION SCHEMES AND THE PROPOSED CVAE MODULATION FILTERING APPROACH.

Cond	MFB	PFB	ETS	RAS	LDA	MHE	CVAE
A. Clean with same Mic							
clean	4.2	4.1	4.5	4.6	4.7	4.0	**3.5**
B: Noisy with same Mic							
Airport	7.5	7.9	8.0	8.1	10.1	8.2	7.1
Babble	7.7	7.9	7.9	8.7	9.9	8.6	7.1
Car	4.7	4.9	5.6	5.0	5.8	4.9	4.3
Rest.	9.8	10.2	11.0	11.0	12.6	11.1	8.9
Street	8.6	8.8	10.0	9.0	10.6	8.8	8.3
Train	8.7	8.3	9.3	9.1	10.6	8.4	8.5
Avg.	7.8	8.0	8.6	8.5	9.9	8.3	**7.4**
C: Clean with diff. Mic							
clean	8.4	7.8	8.0	9.7	10.0	8.1	**6.9**
D: Noisy with diff. Mic							
Airport	19.7	20.9	18.5	20.1	22.3	20.8	18.2
Babble	20.3	20.9	19.3	20.0	22.5	21.3	19.7
Car	11.8	13.1	14.1	12.5	14.5	12.8	10.2
Rest.	21.7	23.7	21.8	23.1	25.2	23.1	19.2
Street	19.1	20.0	19.4	18.9	21.2	20.5	17.2
Train	18.3	19.6	19.6	19.9	21.6	18.9	17.9
Avg.	18.5	19.7	18.8	19.1	21.2	19.6	**17.1**
Avg. of all conditions							
Avg.	12.1	12.7	12.6	12.8	14.4	12.8	**11.2**

TABLE IV
WORD ERROR RATE (%) IN REVERB CHALLENGE DATABASE FOR MULTI-CONDITION TRAINING (SIMULATED) WITH TEST DATA FROM SIMULATED AND REAL REVERBERANT ENVIRONMENTS.

Test Cond	MFB	PFB	RAS	MHE	CVAE
Sim_dev	9.1	8.6	10.1	8.4	**8.3**
Sim_eval	9.5	9.2	10.2	9.2	**8.6**
Real_dev	22.0	21.5	24.9	**21.0**	22.7
Real_eval	25.9	25.9	27.9	**24.5**	24.8
Avg.	16.5	16.3	18.3	**15.8**	16.0

TABLE V
WORD ERROR RATE (%) IN CHIME-3 CHALLENGE DATABASE FOR MULTI-CONDITION TRAINING (REAL+SIMULATED) WITH TEST DATA FROM SIMULATED AND REAL NOISY ENVIRONMENTS.

Test Cond	MFB	PFB	RAS	MHE	CVAE
Sim_dev	14.3	13.7	14.6	14.4	**12.4**
Real_dev	11.6	12.0	11.8	12.0	**10.2**
Avg.	13.0	12.9	13.2	13.2	**11.3**
Sim_eval	25.5	25.1	23.1	26.4	**19.9**
Real_eval	22.6	23.0	21.6	22.9	**18.9**
Avg.	24.1	24.1	22.4	24.7	**19.4**

models in clean and additive noise conditions. For the experiments with different microphone, the ETSI features provide the best performance. The proposed approach performs similar to the PFB features and improve over the baseline MFB features as well.

In Table III for multi-condition training, most of the noise robust front-ends do not improve over the baseline mel-filter bank (MFB) performance (except for condition C), as the acoustic models are trained using multi-condition noisy training data. The proposed CVAE features provide significant improvements in ASR performance over the baseline system (average relative improvements of 7.5%). Furthermore, the improvements in ASR performance are consistently seen across all the noisy conditions of Aurora-4 dataset.

The ASR results on REVERB challenge dataset are shown in Table IV. The proposed approach improves over the baseline features in the REVERB challenge dataset for the simulated conditions and for real reverberation in the evaluation dataset. However, the Hilbert envelope based compensation (MHE) improves over the proposed approach in the evaluation test data for real reverberation.

TABLE VI
WER (%) FOR EACH NOISE CONDITION IN CHIME-3 DATASET WITH THE BASELINE FEATURES AND THE PROPOSED FEATURE EXTRACTION.

Dev Data				
Cond.	Sim		Real	
	MFB	CVAE	MFB	CVAE
BUS	12.6	**10.6**	14.2	**12.6**
CAF	17.0	**15.8**	11.4	**10.2**
PED	12.0	**10.0**	8.5	**7.4**
STR	15.7	**13.2**	12.3	**10.7**
Eval Data				
Cond.	Sim		Real	
	MFB	CVAE	MFB	CVAE
BUS	18.3	**13.8**	29.2	**23.6**
CAF	26.3	**21.4**	23.7	**19.1**
PED	29.1	**21.0**	21.1	**18.6**
STR	28.3	**23.4**	16.4	**14.3**

TABLE VII
STATISTICAL SIGNIFICANCE OF PERFORMANCE IMPROVEMENTS FOR THE PROPOSED METHOD OVER THE BASELINE MFB SYSTEM USING CONFIDENCE INTERVAL AND THE PROBABILITY OF IMPROVEMENT (POI) ON AURORA-4 DATASET. [46].

Test Cond.	Confidence Interval		POI (%)
	MFB	CVAE	
A	[4.0, 5.5]	[3.7, 5.0]	95.0
B	[7.4, 9.7]	[7.1, 9.5]	81.8
C	[7.8, 10.8]	[6.4, 8.9]	100.0
D	[17.5, 23.1]	[16.3, 21.5]	95.3
Avg	–	–	90.0

The results for the CHiME-3 dataset are reported in Table V. The proposed approach to feature extraction provides significant improvements over the baseline system as well as the other noise robust front-ends considered here. On the average, the proposed approach provides relative improvements of 13% in the development set and 20% in the evaluation set. The detailed results on different noises in CHiME-3 are reported in Table VI. For all the noise conditions in CHiME-3 in simulated and real environments, the proposed approach shows significant improvements over the baseline system using mel filter bank features (MFB). In the evaluation dataset, the relative improvements over the baseline features for most of the noise conditions are above 20%.

V. DISCUSSION

A. Comparing various feature extraction methods

For illustrating the benefit of robust feature extraction, the multi-condition ASR experiments are more challenging as the acoustic models are well trained. This may explain the lack of improvements for most of the robust feature extraction front-ends like PFB [8], MHE [9], ETS [44] compared to the baseline features in experiments on the Aurora-4 and CHiME-3 datasets. Even in these matched conditions, the proposed framework of unsupervised filter learning provides significant

TABLE VIII
PERFORMANCE (AVERAGE WER (%) FOR DIFFERENT NUMBER OF MODULATION FILTERS WITHOUT ANY FILTER SELECTION.

No. of 2-D Filters	Aurora-4	REVERB	CHiME-3
2	12.1	**16.3**	**14.9**
3	11.9	16.8	16.4
4	11.6	16.4	16.2
6	**11.5**	16.8	15.9
8	12.0	17.4	16.1

TABLE IX
AVERAGE WER (%) ON ALL THE TEST CONDITIONS OF AURORA-4, REVERB AND CHIME-3 DATASETS.

Mod. Filter	Aurora-4	REVERB	CHiME-3
r_1-s_1, r_1-s_2	13.2	18.0	15.1
r_1-s_1, r_2-s_2	12.1	16.3	**14.9**
r_2-s_1, r_2-s_2	**11.2**	**16.1**	15.3
r_1-s_2, r_2-s_1	12.2	16.4	15.0

improvements compared to the baseline features and other noise robust front-ends.

B. Statistical significance of the ASR results

To compare how one system performs better than other in statistical sense, we use Bootstrap estimate for confidence interval [46]. It computes a bootstrapping of WER to extract the 95% confidence interval (CI), and also gives a probability of improvement (POI) by the system-in-test (system with proposed features) over the reference system (baseline system with MFB features). Table VII shows the analysis for various conditions in the Aurora-4 multi-condition training. The bootstrap estimate of CI is similar for MFB and our proposed CVAE. The POI of CVAE system over the MFB is quite high for almost all test conditions, with average POI being 90%.

C. Choice of number of filters and Filter Selection

The ASR results till now in the paper are reported with only 2 separable modulation filters with filter size as 5×5. In this subsection, we analyze the the effect of different number of filters on the ASR performance without any filter selection (the 2-D filters obtained from the CVAE model are applied directly). These results are reported in Table VIII. From the ASR results, it can be observed that the ASR results do not improve for two of the three datasets considered when the number of modulation filters is increased beyond 2. Hence, we have used only 2 modulation filters in all the other ASR experiments reported in this work.

In the previous section on ASR experiments, we have used the 2-D filters based on $r_2 - s_1$ and $r_2 - s_2$ combinations. While this was partly motivated by the previous studies on human perception of modulation [11], [14], we validate this choice with a set of ASR experiments on multi-condition Aurora-4, REVERB and CHiME-3 datasets. The ASR results using all the four combinations of rate ($r_1; r_2$) and scale filters ($s_1; s_2$) on the databases are shown in Table IX. As seen here, the ASR performance in these experiments validate the claim that the important modulations for ASR lie in the bandpass region of temporal domain and the entire modulation range of the spectral domain, much similar to the human perceptual

TABLE X
EMPIRICAL ANALYSIS ON THE TRAINING MSE LOSS WITH DIFFERENT CVAE NETWORK PARAMETERS.

Activation function			
	ReLU	Tanh	Sigmoid
Loss	39.5	**38.6**	39.3
No. of hidden layers			
	1	2	3
Loss	39.1	**38.6**	39.3
No. of nodes in latent layer			
	5000	3000	512
Loss	38.6	**35.8**	38.0

experiments [11]. In REVERB dataset also, a similar trend is observed. In the CHiME-3 dataset, we found both rate filters r_1 and r_2 to have band pass frequency responses with slightly different bandpass regions. In the ASR experiments, inclusion of rate filter r_2 instead of r_1 gave a degradation in performance. Since the ASR results improved for two out of the three datasets using the proposed filter selection criterion, we have used the same filter selection approach in all other ASR experiments.

D. Choice of network parameters

The chosen parameters of the proposed CVAE architecture is validated by the L2 loss during CVAE training. Table X shows the effect of different parameters such as activation function, number of hidden layers and the number of nodes in the latent layer on the MSE training loss. For each case, the rest of the parameters are kept the same as mentioned in Table I to provide a meaningful comparison. It is seen that the $Tanh$ activation function results in least MSE loss. By varying the number of hidden layers, 2 hidden layers provide the least loss among all. Hence, these parameters have been fixed for training the CVAE model as listed in Table I. In all these experiments, the mini-batch size is chosen as 1200 based on the GPU memory constraints and the computational time needed for the filter learning process.

E. Full-rank vs. Rank-1 filter learning

We compare the proposed feature learning approach using rank-1 constraint with features obtained from unconstrained joint 2-D CVAE modulation filters in Table XI. The full-rank 2-D filters are learnt using the similar cost function as in Eq. 3 except that separable rank-1 constraint is now removed. From the results, we observe that the full-rank features perform worse than rank-1 filters. It is interesting to note that the concept of separable modulation filters has also been observed in ferret auditory cortex [47]. We also find that the feature-level fusion of full-rank and rank-1 filters do not yield any ASR improvements while the score-level fusion yields minor improvements. However, the score-level fusion is computationally expensive as one needs to train two separate ASR systems on each of the feature streams.

F. Choice of generative model loss function for filter learning

All the ASR experiments reported thus far use the filter learning paradigm of the CVAE model with the total loss function defined in Eq. 3. In this subsection, we tease apart

TABLE XI
ASR PERFORMANCE OF PROPOSED 2-D RANK-1 MODULATION FILTERS AND 2-D FULL-RANK JOINT MODULATION FILTERS.

Filter Learning Constraint (WER in %)	
Full rank	12.3
Rank-1 (with filter selection)	11.2
Fusion (Feat.) Full-rank + Rank-1	11.2
Fusion (Score) Full-rank + Rank-1	**11.0**

TABLE XII
EFFECT OF DIFFERENT FILTER LEARNING METHODS ON THE AURORA-4 ASR EXPERIMENTS IN TERMS OF WER.

Discriminative model	WER (%)
CNN (supervised learning of filters)	11.3
Generative model	**WER (%)**
CRBM [27]	12.2
CAE (Only MSE loss in Eq. 3) [48]	11.8
GAN (MSE loss + Adversarial loss) [48]	11.6
Plain CVAE (Only MSE and Latent Loss in Eq. 3)	11.6
Prop. CVAE (All four terms in loss function of Eq. 3)	**11.2**

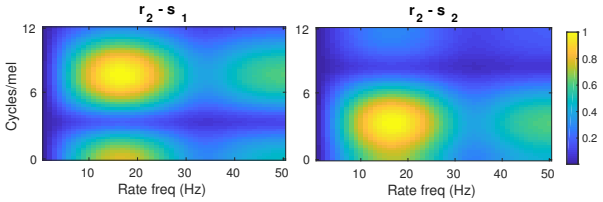

Fig. 4. Two 2-D filters ($\boldsymbol{r}_2 - \boldsymbol{s}_1$) and ($\boldsymbol{r}_2 - \boldsymbol{s}_2$) used in feature extraction for ASR learned from the REVERB Challenge database (8 channels) in CVAE.

the various components of the loss function and analyze their impact for ASR performance on the Aurora-4 task. Specifically, we learn the filters using the conventional CAE model (having only the MSE loss function) as well as the vanilla CVAE model (without the L2 convolution loss or the L1 sparsity loss and having equal weight for the latent loss and the MSE loss). We also compare the proposed CVAE framework for filter learning with the previously proposed convolutional RBM (CRBM) based approach [27] and generative adversarial network (GAN) based approach [48]. All the models are trained in the same framework for learning two filters and use the same training dataset. These ASR experiments on Aurora-4 are reported in Table XII. In addition, we compare these generative model approaches with a discriminative model CNN, where 64 filters in a convolution layer are learned jointly with 4-layer DNN for the ASR task. The results indicate that the generative modeling framework of CVAE (Eq. 3) provides the best ASR performance in comparison with other choices and it improves over the previously proposed CRBM and GAN framework [27], [48]. Also, the proposed approach performs marginally better than the supervised CNN approach of learning filters. The features learned from unsupervised model could also be used in the CNN framework to further improve the ASR performance.

G. Domain specific versus cross-domain filter learning

In Figure 3, the 2-D frequency response of the filters used for feature extraction learned from multi-condition Aurora-4 dataset is shown. The response of the 2-D filters learned

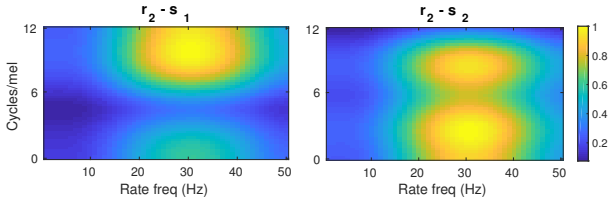

Fig. 5. Two 2-D filters ($\boldsymbol{r}_2 - \boldsymbol{s}_1$) and ($\boldsymbol{r}_2 - \boldsymbol{s}_2$) used in feature extraction for ASR with CHiME-3 database.

TABLE XIII
WER (%) FOR CROSS-DOMAIN ASR EXPERIMENTS.

Filters Learned on	**ASR Trained and Tested on**		
	Aurora-4	REVERB	CHiME-3
Aurora-4	11.2	16.2	15.0
REVERB	11.0	16.1	15.2
CHiME-3	11.2	16.1	15.3

from REVERB challenge and CHiME-3 dataset are shown in Figure 4 and Figure 5 respectively. Comparing the frequency response of the filters learned from each of these datasets, it is observed that the rate filter $\boldsymbol{r}_2$ has relatively lower frequency range in Aurora-4 compared to the other two datasets. In the case of scale filter characteristics, the filters learned from CHiME-3 dataset show higher scale frequency range.

In a subsequent analysis, we perform a cross-domain ASR experiment, i.e., we learn the filters from one of the datasets (either Aurora-4, REVERB Challenge or CHiME-3) and use those filters to train/test ASR on the other two datasets. The results of these cross-domain filter learning experiments are reported in Table XIII. The rows in the table show database used to learn filters and the columns show the dataset used to train and test the ASR. The performance reported in this table are the average WER on each of the datasets. The results shown in Table XIII illustrate that the filter learning process is relatively robust to the domain of the training data used in the CVAE model. This is a key result and it suggests that ASR system is not affected by the minor changes in the filter characteristics observed in Figure 3, 4 and 5. One could assume that the spectro-temporal modulations in noisy/reverberant speech to be composed of components from clean speech and those from noise/reverberation. The experiments in Table XIII lead us to hypothesize that the proposed CVAE based generative model is able to effectively capture the key speech modulations and ignore the spectro-temporal modulations of noise/reverberation. We also hypothesize that the filters learned using Aurora-4 use more training conditions like 6 different noisy conditions and 2 microphone conditions compared to variabilities in the CHiME-3 dataset. Hence, Aurora-4 filters perform the best for CHiME-3. Using the filters learnt from REVERB dataset, the performance on Aurora-4 is improved mainly due to improvements in microphone mis-match condition D.

H. Semi-supervised ASR training

In addition to the ASR experiments with full training data, we also consider the case when only a fraction of the available training data is labeled. This is partly motivated by the fact

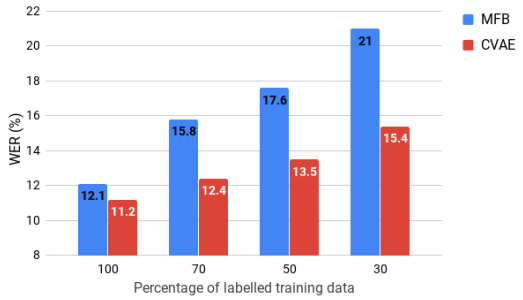

Fig. 6. ASR performance in terms of WER (%) in Aurora-4 database (average of 14 test conditions) for multi condition training using lesser amount of labeled training data (70%, 50%, 30%). Here 100% corresponds to 14 h of training data.

that, while data collection in real noisy environments may be relatively easy, the labeling of noisy data is cumbersome and more expensive than in clean recording conditions. Given the unsupervised learning paradigm of the proposed approach, the 2-D filters could be learned from the entire unlabeled training data and applied for ASR training with the labeled data. We report the ASR experiments with reduced labeled data (70, 50 and 30% random selection of the original training data). These experiments are shown in Figure 6.

It can be observed that the baseline ASR system has a drastic degradation in performance when the amount of training data is reduced. The proposed features using the CVAE model are more resilient to the presence of reduced amounts of labelled training data (a relative improvement of 25% over the baseline for the case with 30% labelled training data). For example, even with 30% of labeled training data, the CVAE feature based ASR attains a WER that is better than the baseline ASR system with 70% labeled training data.

VI. Summary

In this paper, we have proposed a novel framework for modulation filter learning using the convolutional variational autoencoder (CVAE) model. We have presented the mathematical details of the variational model and its application to modulation filter learning. With several speech recognition experiments in a multi-condition training setup, we have also illustrated the performance benefits of the proposed approach compared to baseline methods as well as several other noise robust front-ends proposed in the past.

The key contributions from the paper can be summarized as follows.

Posing the modulation filter learning problem in a generative modeling framework using the convolutional variational autoencoder (CVAE) neural network.

Using modified cost function in the CVAE framework that encourages the filters learned to be irredundant and the latent representation to be sparse. The additional constraints are differentiable and can be easily integrated in the backpropagation learning.

ASR experiments on a variety of speech datasets containing noise and reverberation showing the benefits of the proposed approach.

Exploring the presence of universal modulation characteristics which can be learned from any one of the corpus and generalized to other datasets. The proposed filter learning approach is effective in focusing primarily on the speech modulations in the spectro-temporal domain and ignoring the modulations induced by noise/reverberation regardless of the dataset used for filter learning.

Illustrating the benefits of unsupervised filter learning for semi-supervised ASR applications. The generative modeling framework proposed in this work is able to utilize the unlabeled data effectively for filter learning and reduce the requirement of labeled training data for ASR in noisy conditions.

References

[1] Geoffrey Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[4] Vijayaditya Peddinti, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[5] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7398–7402.

[6] Arun Narayanan and DeLiang Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7092–7096.

[7] Steven Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[8] Chanwoo Kim and Richard M Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4101–4104.

[9] Seyed Omid Sadjadi, Taufiq Hasan, and John HL Hansen, "Mean Hilbert envelope coefficients (MHEC) for robust speaker recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[10] Tomohiro Nakatani, Takuya Yoshioka, Keisuke Kinoshita, Masato Miyoshi, and Biing-Hwang Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[11] Taffeta M Elliott and Frédéric E Theunissen, "The modulation transfer function for speech intelligibility," *PLoS computational biology*, vol. 5, no. 3, pp. e1000302, 2009.

[12] Robert V Shannon, Fan-Gang Zeng, Vivek Kamath, John Wygonski, and Michael Ekelid, "Speech recognition with primarily temporal cues," *Science*, vol. 270, no. 5234, pp. 303–304, 1995.

[13] Rob Drullman, Joost M Festen, and Reinier Plomp, "Effect of temporal envelope smearing on speech reception," *The Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1053–1064, 1994.

[14] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[15] Nima Mesgarani, Malcolm Slaney, and Shihab A Shamma, "Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 920–930, 2006.

[16] Sridhar Krishna Nemala, Kailash Patil, and Mounya Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 2, pp. 416–426, 2013.