ARTIFICIAL BANDWIDTH EXTENSION USING CONDITIONAL VARIATIONAL AUTO-ENCODERS AND ADVERSARIAL LEARNING

Pramod Bachhav, Massimiliano Todisco and Nicholas Evans

EURECOM, Sophia Antipolis, France

{bachhav,todisco,evans}@eurecom.fr

ABSTRACT

Artificial bandwidth extension (ABE) algorithms have been developed to estimate missing highband frequency components (4-8kHz) to improve quality of narrowband (0-4kHz) telephone calls. Most ABE solutions employ deep neural networks (DNNs) due to their well-known ability to model highly complex, non-linear relationship between narrowband and highband features. Generative models such as conditional variational auto-encoders (CVAEs) are capable of modelling complex data distributions via latent representation learning. This paper reports their application to ABE. CVAEs, form of directed, graphical models, are exploited to model the probability distribution of highband features conditioned on narrowband features. While CVAEs are trained with the standard mean square criterion (MSE), their combination with adversarial learning give further improvements. When compared to results obtained with the baseline approach, the wideband PESQ is improved significantly by 0.21 points. The performance is also compared on an automatic speech recognition (ASR) task on the TIMIT dataset where word error rate (WER) is decreased by an absolute value of 0.3%.

Index Terms— variational auto-encoder, generative adversarial network, latent variable, artificial bandwidth extension, speech quality

1. INTRODUCTION

Legacy narrowband (NB) networks and devices typically support bandwidths of 0-4kHz. Today's wideband (WB) networks support bandwidths of 50Hz-8kHz and thus provide improved speech quality. While the transition from NB to WB networks will require significant investments and time [1], artificial bandwidth extension (ABE) algorithms have been developed to improve speech quality when WB devices are used with NB devices or infrastructure. ABE methods estimate missing highband (HB) frequency components above 4kHz from available NB components, typically using a regression model learned from WB training data.

ABE algorithms use either a classical source-filter model [2, 3] or operate directly on complex short-term spectral estimates [4, 5]. Estimation is usually performed via a regression approach using conventional Gaussian mixture [2, 6, 7] and hidden Markov models [8, 9]. Several other approaches exploit superiority of deep neural networks (DNNs) to model non-linear relationship between NB and HB components using Gaussian Bernoulli restricted boltzmann machines (GBRBMs), deep recurrent-neural networks (RNNs) with Long short-term memory (LSTM) cells [10, 11], recurrent temporal restricted Boltzmann machines (RTRBMs) [12]. Some approaches perform ABE via direct modelling and generation of time-domain waveforms [13, 14].

Probabilistic deep generative models such as variational auto-encoders (VAEs) and their conditional variant (CVAEs) are capable of modeling complex data distributions. In contrast to bottleneck features learned by stacked auto-encoders (SAEs), the latent representation is probabilistic and can be used to generate new data. Inspired by their successful use in image processing [15, 16], they have become increasingly popular in numerous fields of speech processing (e.g., speech modelling and transformation [17], voice conversion [18]) and neural machine translation (NMT) [19, 20]. The performance of CVAEs can further be improved by their combination with generative adversarial networks (GANs) [21]. Despite their capability of modelling data distributions, CVAEs have not been investigated in regression tasks such as ABE.

Inspired by the approaches presented in [21,22] for image generation task, the work reported in this paper aims to explore the use of generative modelling techniques to further improve performance of a baseline DNN. In particular, we exploit CVAEs to model distribution of HB features where the conditioning variable of the CVAE is derived from NB features via an auxillary neural network. The performance of the proposed CVAE architecture is further improved via its combination with a GAN. The novel contributions of this work are; (i) the application of CVAEs to ABE for estimation of missing HB features from available NB features; (ii) the combination of CVAE with a probabilistic encoder in the form of an auxillary neural network and their joint optimisation; (iii) adversarial training of the proposed CVAE architecture to further improve the ABE performance.

The remainder of this paper is organised as follows. Section 2 describes a baseline ABE algorithm. Section 3 explains the proposed CVAE scheme and its combination with GAN for ABE. Experimental setup and results are described in Section 4 and conclusions are presented in Section 5.

2. BASELINE ABE ALGORITHM

Fig. 1 illustrates the baseline ABE system. It is identical to the source-filter model based approach presented in [23]. The algorithm is described in *brief* in two blocks: estimation and resynthesis.

During **estimation**, a NB speech frame $\mathbf{s}^{\mathrm{NB}}$ of 30 ms duration with a sampling rate of 16kHz is processed using a 512-point FFT in order to extract 128-dimensional NB log power spectrum ($\mathrm{LPS}_{\mathrm{NB}}$) coefficients $\mathbf{x}^{\mathrm{NB}}$. Mean and variance normalisation ($\mathrm{mvn}_{\mathbf{x}}$) is then applied to obtain $\mathbf{x}^{\mathrm{NB}}_{\mathrm{mvn}}$. After concatenation with the coefficients obtained from 2 neighbouring frames, the resulting 640-dimensional concatenated vector $\mathbf{x}^{\mathrm{NB}}_{\mathrm{conc\_2}}$ is then fed to a DNN to estimate 10-dimensional normalised HB features $\hat{\mathbf{y}}^{\mathrm{HB}}_{\mathrm{mvn}}$ consisting of first 10 linear prediction cepstral coefficients (LPCCs). Inverse mean and variance normalisation ($\mathrm{mvn}_{\mathbf{y}}^{-1}$) is then applied, giving HB features $\hat{\mathbf{y}}^{\mathrm{HB}}$. The

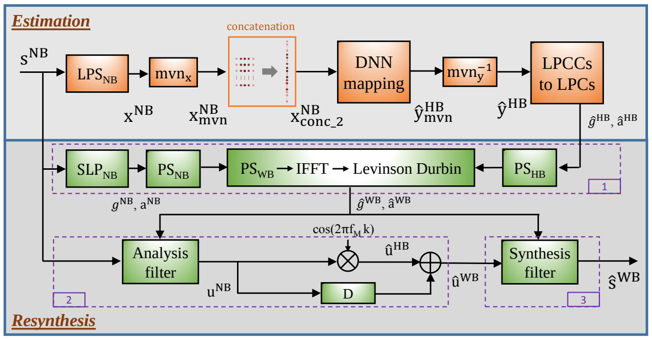

Fig. 1. *A block diagram of the baseline ABE system. Diagram adapted from [23].*

HB LP coefficients $\hat{g}^{\text{HB}}, \hat{\mathbf{a}}^{\text{HB}}$ are then calculated from the estimated HB LPCCs ($\mathbf{y}^{\text{HB}}$) via recursion.

Resynthesis is performed in three steps. First (box in Fig. 1), LP parameters $\mathbf{a}^{\text{NB}}$, g^{NB} are obtained from speech frame $\mathbf{s}^{\text{NB}}$ via selective linear prediction (SLP_{NB}) to get the NB power spectrum PS_{NB}. This is then concatenated with the HB power spectrum PS_{HB} (obtained from estimated HB LP parameters $\hat{g}^{\text{HB}}, \hat{\mathbf{a}}^{\text{HB}}$), giving the WB power spectrum PS_{WB}, and hence estimated WB LP parameters $\hat{g}^{\text{WB}}, \hat{\mathbf{a}}^{\text{WB}}$. Second (box 2), the HB excitation $\hat{\mathbf{u}}^{\text{HB}}$ is estimated from the spectral translation of the NB excitation $\mathbf{u}^{\text{NB}}$ with $f_{\text{M}} = 8$ kHz (which corresponds to spectral folding around 4kHz). NB and HB excitation components are then combined to obtain the extended WB excitation $\hat{\mathbf{u}}^{\text{WB}}$. Finally (box 3), $\hat{\mathbf{u}}^{\text{WB}}$ is filtered using a synthesis filter defined by $\hat{g}^{\text{WB}}$ and $\hat{\mathbf{a}}^{\text{WB}}$ in order to resynthesise speech frame $\hat{\mathbf{s}}^{\text{WB}}$. A conventional overlap and add (OLA) technique is used to produce extended WB speech.

3. APPLICATION OF CVAE AND GAN FOR ABE

In this section we describe how CVAEs can be used for estimation of HB features from input NB features in an ABE task. The CVAEs are trained in an adversarial fashion to deliver improvements in ABE performance.

3.1. Conditional variational auto-encoders

A conditional variational auto-encoder (CVAE) is a conditional, generative model of the form $p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z})p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})$. For a given input observation $\mathbf{x}$, a latent variable $\mathbf{z}$ is drawn from a prior distribution $p_\theta(\mathbf{x})$ from which the posterior distribution $p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})$ generates the output $\mathbf{y}$ [15, 16].

CVAE maximises the conditional likelihood $p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})$ via the use of recognition/inference model $q_\phi(\mathbf{z}|\mathbf{y})$[1] (also referred to as probabilistic encoder) which estimates the parameters ϕ of the posterior distribution over all possible values of the latent variables $\mathbf{z}$ that may have generated the given datapoint $\mathbf{y}$. The probabilistic decoder $p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})$ then produces a distribution with parameters θ over all possible values of $\mathbf{y}$ for given $\mathbf{z}$ and $\mathbf{x}$. For simplicity, it is assumed that the approximate ($q_\phi(\mathbf{z}|\mathbf{y})$) and true posteriors ($p_\theta(\mathbf{z}|\mathbf{y})$) are diagonal multivariate Gaussian distributions whose respective parameters ϕ and θ are computed using two different DNNs.

[1] In our formulation, we assume that the latent variable $\mathbf{z}$ is dependent only on the output variable $\mathbf{y}$ i.e., $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{y})$

The variational lower bound on the conditional likelihood $p_\theta(\mathbf{y}|\mathbf{x})$ is then given by:

$$\begin{aligned}\log p_\theta(\mathbf{y}|\mathbf{x}) &\geq \mathcal{L}(\theta,\phi;\mathbf{x},\mathbf{y}) \\ &= -D_{KL}[q_\phi(\mathbf{z}|\mathbf{y})||p_\theta(\mathbf{z})] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})] \end{aligned} \quad (1)$$

where $D_{KL}(\cdot)$ acts as a regulariser which can be computed analytically. In practice, the prior $p(\mathbf{z})$ is assumed to be a centred isotropic multivariate Gaussian $\mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$ with no free parameters. The second term is approximated via sampling by $\frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}^{(l)})$ using L samples drawn from the recognition network $q_\phi(\mathbf{z}|\mathbf{y})$. Sampling is performed using a differentiable deterministic mapping such that $\mathbf{z}^{(l)} = g_\phi(\mathbf{y},\epsilon^{(l)}) = \boldsymbol{\mu}_\mathbf{z} + \epsilon^{(l)} \odot \boldsymbol{\sigma}_\mathbf{z}$ where $\epsilon^{(l)} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. $\boldsymbol{\mu}_\mathbf{z} = \boldsymbol{\mu}(\mathbf{y};\phi)$ and $\boldsymbol{\sigma}_\mathbf{z} = \boldsymbol{\sigma}(\mathbf{y};\phi)$ are outputs of the recognition network $q_\phi(\mathbf{z}|\mathbf{y})$. This is called the *reparameterization trick*.

The output distribution $p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})$ in Eq. 1 is chosen to be Gaussian with mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance matrix $\sigma^2 * \mathbf{I}$, i.e., $p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \sigma^2 * \mathbf{I})$ where $\boldsymbol{\mu}_\mathbf{x}$ is output of the decoder DNN $\boldsymbol{\mu}(\mathbf{x},\mathbf{z};\theta)$. Therefore second term in Eq. 1 can be re-written as:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})] = C - \frac{1}{L}\sum_{l=1}^{L}\frac{\|\mathbf{y} - \boldsymbol{\mu}(\mathbf{x},\mathbf{z}^{(l)};\theta)\|^2}{\alpha} \quad (2)$$

where C is a constant that can be ignored during optimisation. The scalar $\alpha = 2\sigma^2$ can be seen as a weighting factor between the KL-divergence and the reconstruction term. In practice, $L = 1$ samples are used per datapoint [24]. The lower bound $\mathcal{L}(\cdot)$ forms the objective function which can be optimized with respect to parameters θ and ϕ using a stochastic gradient descent algorithm.

3.2. Generative adversarial networks

In regression framework, a GAN consists of two adversarial networks, a generator G and a discriminator D. Generator maps an input sample $\mathbf{x}$ to an output sample $\mathbf{y}$. D takes form of a binary classifier which predicts the probability $D(\mathbf{x})$ that a given sample $\mathbf{x}$ belongs to the training distribution and not the distribution modelled by G. D is thus trained to maximise the probability of assigning a correct label to both training samples and samples from G [25]. This adversarial learning process is formulated as a minimax game between G and D given according to the following objective function:

$$\min_G \max_D V(G,D) = \mathbb{E}_\mathbf{y}[\log(D(\mathbf{x}))] + \mathbb{E}_\mathbf{x}[\log(1 - D(G(\mathbf{x})))] \quad (3)$$

During training, G tries to fool D by generating samples close to the training data so that D classifies G's output as *real*. D then updates its parameters in order to classify samples generated by G as *fake*. Both G and D are trained iteratively until the GAN converges to a good estimator of the true data distribution [25].

3.3. Application to ABE

This section describes the proposed scheme for optimisation of CVAEs specifically tailored to ABE. The scheme is illustrated in Fig. 2(a). Parallel training data consisting of NB and WB utterances is processed in frames of 30ms duration with 15ms overlap. Input data $\mathbf{x} = \mathbf{x}^{\text{NB}}_{\text{conc\_2}}$ consists of NB LPS coefficients with memory (as described in Section 2). The output data $\mathbf{y} = \mathbf{y}^{\text{HB}}_{\text{mvn}}$ consists of first 10 LPCCs extracted from parallel HB data via SLP.

The CVAE is then trained to model the distribution of the output $\mathbf{y}$ conditioned on the input $\mathbf{x}$ as follows. The HB data $\mathbf{y}$ is fed

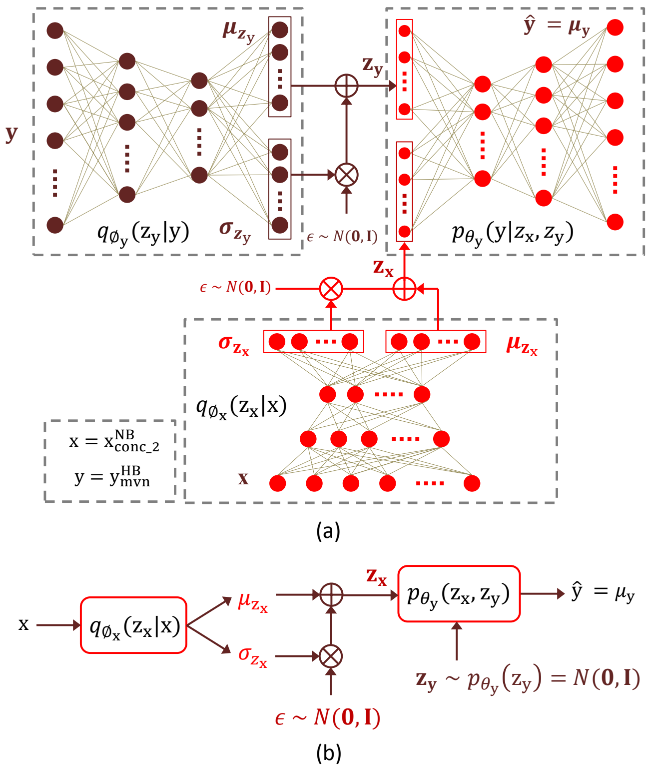

Fig. 2. *(a) The proposed CVAE-DNN scheme during training (or reconstruction) phase and (b) testing (or prediction) phase.*

to the encoder $q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})$ (top-left network in Fig. 2(a)) in order to predict the mean $\boldsymbol{\mu}_{\mathbf{z_y}}$ and log-variance $\log(\boldsymbol{\sigma}^2_{\mathbf{z_y}})$ of the approximate posterior distribution $q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})$. The predicted parameters are then used to obtain the latent representation $\mathbf{z_y} \sim q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})$ of the output variable $\mathbf{y}$ via the *reparameterization trick* (see Section 3.1). The encoder $q_{\phi_\mathbf{x}}(\mathbf{z_x}|\mathbf{x})$ (bottom of Fig. 2(a)) is fed with input data $\mathbf{x}$ in order to predict the mean $\boldsymbol{\mu}_{\mathbf{z_x}}$ and log-variance $\log(\boldsymbol{\sigma}^2_{\mathbf{z_x}})$ that represent the posterior distribution $q_{\phi_\mathbf{x}}(\mathbf{z_x}|\mathbf{x})$. The latent variable $\mathbf{z_x} \sim q_{\phi_\mathbf{x}}(\mathbf{z_x}|\mathbf{x})$ is then used as the CVAE conditioning variable.

After concatenation, $\mathbf{z_x}$ and $\mathbf{z_y}$ are fed to the decoder $p_{\theta_\mathbf{y}}(\mathbf{y}|\mathbf{z_x}, \mathbf{z_y})$ (top-right network) in order to predict the mean $\boldsymbol{\mu}_\mathbf{y} = \boldsymbol{\mu}(\mathbf{z_x}, \mathbf{z_y}; \theta)$ of the output variable $\mathbf{y}$. Finally, the entire network is trained to learn parameters $\phi_\mathbf{x}$, $\phi_\mathbf{y}$ and $\theta_\mathbf{y}$ jointly. From Eqs. 1 and 2, the equivalent variational lower bound under optimisation is given by:

$$\log p_{\theta_\mathbf{y}}(\mathbf{y}|\mathbf{z_x}) \geq \mathcal{L}(\theta_\mathbf{y}, \phi_\mathbf{y}, \phi_\mathbf{x}; \mathbf{z_x}, \mathbf{y}) =$$
$$-\left[D_{KL}[q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})||p_{\theta_\mathbf{y}}(\mathbf{z_y})] - \frac{1}{L}\sum_{l=1}^{L} \frac{\|\mathbf{y} - \boldsymbol{\mu}(\mathbf{z_x}, \mathbf{z_y}^{(l)}; \theta_\mathbf{y})\|^2}{\alpha}\right] \tag{4}$$

It is expected that, during optimisation of Eq. 4, parameters $\phi_\mathbf{x}$, $\phi_\mathbf{y}$ and $\theta_\mathbf{y}$ are jointly updated so that the framework learns to *reconstruct* HB data $\mathbf{y}$ from the input data $\mathbf{x}$.

Finally after training (i.e. *reconstruction* phase), the encoder $q_{\phi_\mathbf{x}}(\mathbf{z_x}|\mathbf{x})$ and the decoder $p_{\theta_\mathbf{y}}(\mathbf{y}|\mathbf{z_x}, \mathbf{z_y})$ (signified by the red components in Fig. 2(a)) networks are used to form a DNN (denoted as CVAE-DNN) with two stochastic layers $\mathbf{z_x}$ and $\mathbf{z_y}$. The proposed CVAE-DNN scheme is illustrated in Fig. 2(b). It can be used

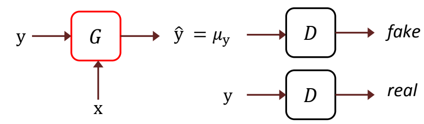

Fig. 3. *An illustration of an adversarial training where G network is formed using CVAE architecture shown in Fig. 2(a).*

for estimation of $\mathbf{y}$ where $\mathbf{z_y}$ is sampled from the prior distribution $p_{\theta_\mathbf{y}}(\mathbf{z_y}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ during testing (or *estimation* phase). It can be noted that there exists a discrepancy during *reconstruction* and *estimation* phases of CVAEs since $\mathbf{y}$ is not available during estimation [20].

3.4. Combining CVAE with GAN

In ABE framework, the reconstruction error term in the objective function given in Eq. 4 can be replaced by GAN based objective loss function (described in Section 3.2) [1, 26]. In this work, this is achieved by forming a generator network G using the proposed CVAE scheme (which is described in Section 3.3 and shown in Fig. 2(a)). As shown in Fig. 3, G then trained to reconstruct $\hat{\mathbf{y}}$ from the inputs $\mathbf{x}$ and $\mathbf{y}$. D is then optimised to classify the generated ($\hat{\mathbf{y}}$) and real ($\mathbf{y}$) samples correctly. Both G and D are trained simultaneously. The objective function of the CVAE-GAN scheme is thus given according to:

$$\mathcal{L}(\theta_\mathbf{y}, \phi_\mathbf{y}, \phi_\mathbf{x}; \mathbf{z_x}, \mathbf{y}) = -D_{KL}[q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})||p_{\theta_\mathbf{y}}(\mathbf{z_y})]$$
$$- \left[\min_G \max_D V(G, D) + \lambda\mathcal{L}_1(G)\right] \tag{5}$$

The error term $\mathcal{L}_1(G) = \mathbb{E}_\mathbf{y}\|\mathbf{y} - \hat{\mathbf{y}}\|_1$ is added in order to minimise the distance between generated ($\hat{\mathbf{y}}$) and real ($\mathbf{y}$) samples, the trick that helps to stabilise GAN training [27]. After adversarial training of the CVAE, a DNN (denoted as CVAE-GAN) is formed from G network in a similar fashion as described in 3.3 and shown in Fig. 2(b)).

4. EXPERIMENTAL SETUP AND RESULTS

This section describes the databases used for ABE experiments, baseline algorithm, configuration details of CVAE and GAN architectures, and results. Experiments are designed to compare the performance of ABE systems that use CVAE-DNN and CVAE-GAN with that uses a DNN trained with conventional MSE criterion.

4.1. Database

The dataset by Valentini et al. [28] was used for training and validation. The training set consists of 11572 sentences spoken by 28 English speakers at a sampling rate of 48kHz. Parallel NB and WB speech signals were created by downsampling the original files to 8 and 16kHz respectively. While 80% of feature vectors extracted from the training set were used for training DNN models, remaining 20% samples were used for validation. The acoustically-different TIMIT core test subset [29] was used for testing.

4.2. CVAE configuration and training

The CVAE architecture[2] is implemented using the Keras toolkit [30]. Encoders $q_{\phi_\mathbf{x}}(\mathbf{z_x}|\mathbf{x})$ and $q_{\phi_\mathbf{y}}(\mathbf{z_y}|\mathbf{y})$ consist of one hidden layer with

[2]Implementation is available at `https://github.com/bachhavpramod/bandwidth_extension`

Table 1. *Objective assessment results. Lower values of RMS-LSD (in dB) reflect better performance whereas higher values of segSNR (in dB) and MOS-LQO$_{WB}$ values indicate better performance.*

Regression method	RMS-LSD	segSNR	WB-PESQ
DNN	**8.90**	17.09	3.35
CVAE-DNN	9.11	17.54	3.41
CVAE-GAN	10.21	**17.81**	**3.56**

Table 2. *ASR WER performance in % on the TIMIT core test subset.*

Regression method	mono	tri1	tri2	tri3
up-NB	39.4	35.7	31.9	27.2
DNN	35.4	29.3	26.2	24.4
CVAE-DNN	35.1	29.2	**25.9**	24.3
CVAE-GAN	**34.5**	**28.8**	26.1	**24.1**
WB	32.2	25.5	23.7	21.3

128 units, and 640 and 10 units for input layers respectively. Their outputs are Gaussian-distributed latent variable layers $\mathbf{z_x}$ and $\mathbf{z_y}$ consisting of 64 units for the means $\boldsymbol{\mu}_{\mathbf{z_x}}$, $\boldsymbol{\mu}_{\mathbf{z_y}}$ and log-variances $\boldsymbol{\sigma}_{\mathbf{z_x}}$, $\boldsymbol{\sigma}_{\mathbf{z_y}}$. The decoder $p_{\theta_\mathbf{y}}(\mathbf{y}|\mathbf{z_x}, \mathbf{z_y})$ consist of one hidden layer with 128 units and an output layer with 10 units respectively. All hidden layers have *tanh* activation units whereas Gaussian parameter layers have *linear* activation units. The modelling of log-variances avoids the estimation of negative variances.

Training is performed in order to minimise the negative conditional log-likelihood in Eq. 4 using the Adam stochastic optimisation technique [31] with an initial learning rate of 10^{-3} and hyperparameters $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The hyperparameter α (in Eq. 4) is set to 10 (i.e., dimension of HB feature vectors $\mathbf{y}$) according to our previous investigations reported in [32]. Networks are initialised according to the approach described in [33] so as to improve the rate of convergence. To discourage overfitting, batch-normalisation [34] is applied before every activation layer. The learning rate is reduced by half when the validation loss increases between 5 consecutive epochs. The CVAE is trained for a 50 epochs using input $\mathbf{x}$ and output $\mathbf{y}$ data with a batch size of 512. The model giving the lowest validation loss is used for subsequent processing.

During adversarial training, the generator G network comprises the same CVAE architecture described above. The discriminator D is a binary classifier with *sigmoid* activation unit and it consists of three hidden layers with 512 *tanh* activation units. The hyperparameter λ (in Eq. 5) is set to 1. The GAN is trained for 30000 iterations with a batch size of 512 using same optimization technique and learning rate.

The performance of the proposed schemes is compared with a baseline DNN of same architecture that is trained using a MSE criterion. All networks have a common structure of (128,128,128) hidden units in order to maintain low complexity given that ABE is a real-time application.

4.3. Objective assessment

Objective assessment metrics include root mean square log-spectral distortion (RMS-LSD) (calculated for a frequency range of 4-8kHz), segmental signal-to-noise ratio (segSNR) and a WB extension to the perceptual evaluation of speech quality algorithm (WB-PESQ) [35]. The latter gives objective estimates of mean opinion scores.

Results are presented in Table 1. The proposed approaches (CVAE-DNN and CVAE-GAN) outperform baseline DNN in terms segSNR and WB-PESQ. Surprisingly, the baseline DNN achieves better (lower) RMS-LSD value than the proposed schemes.

4.4. Word error rate (WER) assessments on ASR task

Objective metrics may not provide accurate estimates, subjective listening tests are thus necessary in order to assess ABE performance. However, obtaining reliable quality estimates via subjective tests is time-consuming and expensive. Thus, we evaluated merit of the proposed methods in terms of WER performance for an ASR task.

Four GMM-HMM based ASR systems were trained (using the Kaldi toolkit [36]) on WB speech signals obtained from the TIMIT training set using monophone (mono) and triphone (tri1, tri2 and tri3) modelling schemes. Note that the aim of this experimental setup is to investigate performance of different ABE algorithms using simple ASR systems. The performance of the ASR systems was tested using the core TIMIT test subset. First, all WB test signals were processed to produce upsampled NB (up-NB) signals at a sampling rate of 16kHz. The up-NB signals were then bandwidth-extended using DNN, CVAE-DNN and CVAE-GAN based ABE systems. ASR was then performed for these test signals using a model that was trained on original WB signals.

The WER results are shown in Table 2. Highest WER (indicates lower performance) is obtained for up-NB signals when tested with the ASR model trained on WB data, this is obvious because HB frequency content is missing in up-NB signals. While all ABE approaches achieve lower WERs than up-NB signals, the proposed CVAE based approaches consistently improve ASR performance over the DNN based baseline method. The CVAE-DNN outperforms DNN by WER improvement of 0.3% for tri2 ASR system. CVAE-GAN achieves lower WER by 0.9% (mono), 0.5% (tri1) and 0.3% for remaining ASR systems. The WER assessment thus indicates that the proposed ABE systems produce bandwidth-extended signals closer to original WB signals. Few speech samples are available at `http://audio.eurecom.fr/content/media`.

5. CONCLUSIONS

Conditional variational auto-encoders (CVAEs) are directed graphical models that are used for generative modelling. This paper reports their application to ABE for modelling distribution of highband features. The conditioning variable of the CVAE is derived from narrowband features via an auxillary neural network. CVAEs are also combined with GAN framework via adversarial training in order to seek further improvements. The proposed approaches produce of speech of substantially better quality which is confirmed via improvements in WB-PESQ, segSNR estimates. The merit of the proposed approach is further assessed via improvements in ASR performance. Crucially the improvements are achieved without augmenting complexity of the baseline regression model. Future work should investigate why spectral distance measure did not show correlation with the other performance metrics. Better CVAE training strategies in order to reduce the discrepancy during *reconstruction* and *estimation* phases should bring further improvements to ABE performance.

6. REFERENCES

[1] S. Li, S. Villette, P. Ramadas, and D. Sinder, "Speech bandwidth extension using generative adversarial networks," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5029–5033.

[2] K.-Y. Park and H. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000, pp. 1843–1846.

[3] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, no. 8, pp. 1707–1719, 2003.

[4] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4395–4399.

[5] P. Bachhav, M. Todisco, M. Mossi, C. Beaugeant, and N. Evans, "Artificial bandwidth extension using the constant Q transform," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5550–5554.

[6] A. Nour-Eldin and P. Kabal, "Mel-frequency cepstral coefficient-based bandwidth extension of narrowband speech,," in *Proc. of INTERSPEECH*, 2008, pp. 53–56.

[7] P. Bachhav et al., "Artificial bandwidth extension with memory inclusion using semi-supervised stacked auto-encoders," in *Proc. of INTERSPEECH*, 2018, pp. 1185–1189.

[8] I. Katsir, D. Malah, and I. Cohen, "Evaluation of a speech bandwidth extension algorithm based on vocal tract shape estimation," in *Proc. of Int. Workshop on Acoustic Signal Enhancement (IWAENC)*. VDE, 2012, pp. 1–4.

[9] J. Abel, M. Strake, and T. Fingscheidt, "Artificial bandwidth extension using deep neural networks for spectral envelope estimation," in *Proc. of Int. Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2016, pp. 1–5.

[10] Y. Gu, Z.-H. Ling, and L.-R. Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks.," in *Proc. of INTERSPEECH*, 2016, pp. 297–301.

[11] K. Schmidt and B. Edler, "Blind bandwidth extension based on convolutional and recurrent deep neural networks," in *in Proc. of Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5444–5448.

[12] Y. Wang et al., "Speech bandwidth extension using recurrent temporal restricted boltzmann machines," *IEEE Signal Processing Letters*, pp. 1877–1881, 2016.

[13] V. Kuleshov et al., "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[14] T. Lim, R. Yeh, Y. Xu, M. Do, and M. Hasegawa-Johnson, "Time-frequency networks for audio super-resolution," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 646–650.

[15] K. Sohn et al., "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.

[16] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *European Conference on Computer Vision*. Springer, 2016, pp. 776–791.

[17] M. Blaauw and J. Bonada, "Modeling and transforming speech using variational autoencoders.," in *INTERSPEECH*, 2016, pp. 1770–1774.

[18] C.-C. Hsu et al., "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. of IEEE APSIPA*, 2016, pp. 1–6.

[19] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang, "Variational neural machine translation," *arXiv preprint arXiv:1605.07869*, 2016.

[20] A. Pagnoni, K. Liu, and S. Li, "Conditional variational autoencoder for neural machine translation," *arXiv preprint arXiv:1812.04405*, 2018.

[21] A. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015.

[22] J. Bao et al., "CVAE-GAN: fine-grained image generation through asymmetric training," in *Proc. of the IEEE Int. Conf. on Computer Vision*, 2017, pp. 2745–2754.

[23] P. Bachhav, M. Todisco, and N. Evans, "Exploiting explicit memory inclusion for artificial bandwidth extension," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5459–5463.

[24] D. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[25] I. Goodfellow et al., "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[26] J. Sautter et al., "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7005–7009.

[27] S. Pascual et al., "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[28] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech.," in *SSW*, 2016, pp. 146–152.

[29] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, and D. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report N*, vol. 93, 1993.

[30] F. Chollet et al., "Keras," `https://github.com/keras-team/keras`, 2015.

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

[32] P. Bachhav et al., "Latent representation learning for artificial bandwidth extension using a conditional variational auto-encoder," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7010–7014.

[33] K. He et al., "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE int. conf. on computer vision*, 2015, pp. 1026–1034.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. conf. on machine learning*, 2015, pp. 448–456.

[35] "ITU-T Recommendation P.862.2 : Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs," *ITU*, 2005.

[36] D. Povey et al., "The kaldi speech recognition toolkit," in *IEEE Workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.