IEEE Signal Processing Society
IEEE Transactions on Audio, Speech and Language Processing

End-to-End Adversarial Blind Bandwidth Extension with Convolutional and Recurrent Networks

Journal:	*Transactions on Audio, Speech and Language Processing*
Manuscript ID	T-ASL-08562-2021
Manuscript Type:	Regular Paper
Date Submitted by the Author:	14-Jul-2021
Complete List of Authors:	Schmidt, Konstantin; International Audio Laboratories Erlangen Mahmoud, Ahmed; Fraunhofer IIS, Fuchs, Guillaume; Fraunhofer Institut, Audio Edler, Bernd; International Audio Laboratories Erlangen; Fraunhofer IIS,
EDICS:	SPE-CODI Speech Coding < SPEECH PROCESSING, SPE-ENHA Speech Enhancement and Separation < SPEECH PROCESSING, SPE-ROBU Robust Speech Recognition < SPEECH PROCESSING

End-to-End Adversarial Blind Bandwidth Extension with Convolutional and Recurrent Networks

Konstantin Schmidt, Ahmed Mustafa Mahmoud, Guillaume Fuchs, and Bernd Edler

***Abstract*—Blind bandwidth extension improves the perceived quality and intelligibility of telephone-quality speech by artificially regenerating missing frequency content that is not coded and transmitted by speech codecs. This work proposes novel approaches based on deep neural networks to solve this problem. These are based on convolutional or on recurrent architectures. All operate in time-domain. Motivated by the source-filter model of the human speech production, two of the proposed systems decompose speech signals into spectral envelopes and excitation signals. Each of them are bandwidth extended separately with a dedicated DNN. All systems are trained with a mixture of adversarial and perceptual loss. To avoid mode collapse and a more stable adversarial training, spectral normalisation is employed in the discriminator. The presented systems are compared to previously published systems by objective measures and subjectively by a listening test. An estimation of the computational complexity is given and compared to state of the art speech coding technologies. Objective and subjective tests show that the proposed systems deliver substantial better quality than prior techniques. It was further shown that our systems reduces the Word Error Rate of a speech recognition systems.**

***Index Terms*—bandwidth extension, artificial bandwidth expansion, speech enhancement, audio super resolution, speech super resolution**

I. Introduction

SPEECH communication is a technology used by most people every day, creating a vast amount of data that needs to be transmitted over Voice over Internet Protocol (VoIP), cellular or public switched telephone networks[1]. While the amount of transferred data should be kept low, the quality of speech is desired to be high. In order to reach this goal, speech compression technologies have evolved over the past decades from compressing bandlimited speech with simple pulse code modulation [1] to coding schemes following speech production and human perception models able to code fullband speech [2], [3]. Albeit the existence of such standardised speech codecs, their adoption in cellular or public switched telephone networks takes years if not decades. For this reason AMR-NB [4] remains the most frequently used codec for mobile speech communication which merely encodes frequencies from 200 Hz to 3400 Hz (usually named *narrowband*, NB). However, transmitting band-linited speech not only harms the acoustic quality but also the intelligibility [5], [6], [7]. Blind bandwidth extension (BBWE) - also known as artificial bandwidth expansion or audio super resolution - artificially regenerates missing frequency components without transmitting additional information from the encoder. A BBWE can be added to the decoder toolchain without any adaption of the transmission network and thus can serve as an intermediate solution to improve the perceived audio quality and intelligibility until better codecs will be deployed in the network [5], [6], [8]. For the sake of transmission bandwidth saving or quality improvement, robust BBWE can still be a viable solution for modern speech transmission. In addition and for other types of applications such as audio restoration, where band-limited speech is stored or archived, BBWE is the only possible solution to expand the audio bandwidth.

Despite the fact that BBWE has a long tradition in speech and audio signal processing community, [9], [10] it is only recently that solutions based on deep neural networks (DNN) have been considered if being developed by researchers with a background in artificial intelligence (AI) or image processing, rather than in speech signal processing. Such DNN-based systems are commonly called speech super resolution (SSR). In image processing, the task of estimating a high-resolution image from one or more low-resolution observations is referred as super-resolution and has received substantial attention within the computer vision community. Recently, Deep Convolutional Neural Networks have achieved better results than traditional methods [11], while super-resolution generative adversarial networks are considered as state-of-the-art [12]. Generative Adversarial Networks (GAN) can restitute better the finer structure for a more realistic reproduction. However, some of these systems can not be directly applied to speech communication scenarios. Besides the fact that the underlying signal is of different nature (e.g. of different dimensionality), there are more aspects to be considered in the design of a BBWE: first of all, it is necessary that the algorithmic delay - that is the time the decoded speech lags behind the original speech - is not too large. Furthermore, the computational complexity and memory consumption must satisfy requirements for real-time processing on embedded systems, such as on mobile phones.

Recurrent Neural Networks are well suited for analysing or predicting time-series, like speech. Indeed, speech can be considered as wide-sense stationary or quasi-periodic on durations of about 20 to 25 ms, and its time correlation can be exploited in RNN with relatively small models. On the other hand, CNN are performant in pattern recognition and upscaling tasks, as in image super-resolution. They also have the advantage that processing can be highly parallelised. Therefore, for speech processing, and particularly for BBWE, both architectures deserve to be considered.

This work presents two BBWEs based on deep neural networks using adversarial learning targeting speech coding

Manuscript received July XX, 2021; revised August XX, 2021.

[1]According to a 2017 OFCOM study an average of 156.75 monthly outbound mobile call minutes are made per subscription: https://www.ofcom.org.uk/research-and-data/multi-sector-research/cmr/cmr-2018/interactive

scenarios. To summarise our contribution:

- We propose two novel deep network structures for the purpose of blind bandwidth extension, one based on convolutional kernels and the other one based on recurrent kernels.
- Both networks are trained with a mixture of adversarial and spectral loss.
- To our best knowledge both systems are the first BBWEs trained adversarially and "end-to-end" - meaning that the input is time-domain speech as well as the output.
- We apply hinge loss and spectral normalisation to increase the performance of the GAN.

A good BBWE not only increases the perceived quality of speech but can also improve word error rates of automated speech recognition systems [13].

II. State of the Art

As mentioned before, the principle of BBWE was originally presented by Karl-Otto Schmidt in 1933 [9], using analog nonlinear devices to extend the bandwidth of transmitted speech. The idea of doing (non-blind) bandwidth extension on the excitation signal of speech codecs dates back to at least 1959 [10]. In the following years several so called parametric BWEs were presented that, motivated by the source-filter model of the human speech production, utilised the separation of the speech signal into excitation and spectral envelope. These systems apply statistical models to extrapolate the spectral envelope while generating the excitation signal by spectral folding [14], spectral translation [8] or by nonlinearities [15]. The statistical models for envelope extrapolation are simple codebook mappings [16], hidden Markov models [14], (shallow) neural networks [17], or recently DNNs [18].

Before using DNNs, the input to the statistical models were often hand-tailored features [14], [17], [19], [20]. With the introduction of DNNs, this approach can be simplified to directly using logarithmic short-time Fourier transform (STFT) energies [18], [21], [22] or the time-domain speech signal [23], [24], [25]. The same is true for the output of the statistical models. Instead of modelling sub-band energies [8] or other envelope representations [21], DNNs are powerful enough to model spectral magnitudes per bin [15], if not the whole time-domain speech signal or a combination of time-domain and frequency domain [26]. However, if the spectral magnitude is modelled, the phase still needs to be reconstructed by spectral folding or translation [18], [21], [15], [27].

1) Training Objective: Designing an efficient DNN-based solution requires selecting the appropriate architecture, and primarily a careful choice of the learning loss function and network type. Typical loss functions are: mean-squared error [21], categorical cross entropy (CE) loss [28], adversarial loss [29], [30], [25] or a mixture of losses [31]. The loss function can also determine the data representation.

A. Mean-squared Error and Cross Entropy

MSE loss, in combination with logarithmic sub-band or bin energies allows for a psychoacoustically motivated loss [8]. CE-derived loss functions, on the other hand, predict sample bits (or sample magnitudes) as classes and therefore the signal to be modelled needs to be quantised with not too high resolution to be handled by DNNs. Predicting the 2^{16} classes of a speech signal quantised with 16 bits is still very costly to be handled by DNNs up to the present day. Fortunately, it is sufficient to quantise the speech signal with 8 bits without any noteworthy loss in quality if the signal is preshaped by a nonlinear function [32], [33], [23], [24], [34]. The nonlinear function used here is the $\mu-$law function, the very same used in the first ever standardised digital speech codec [1]:

$$\hat{x} = \mathrm{sgn}(x)\frac{\ln(1+255|x|)}{\ln(256)}, -1 < x < 1. \quad (1)$$

The values of $\hat{x}$ are rounded to integers and coded with 8 bits. Applying this function to values of x enables coarser quantisation of values with high magnitude and finer quantisation of values with smaller magnitude. This is desired since loud signals can better mask larger quantisation errors. Further more, this function makes the Laplacian-like probability distribution of speech samples more Gaussian.

1) Adversarial Loss: The distribution of time-domain speech is very complex and hard to model, even with today's powerful networks. Generative models trained with MSE or CE loss to match this complex distribution, will only produce a smoothed approximation thereof. When applied to BBWE, this means that the resulting speech signal will lack crispness and energy [30].

Generative adversarial networks [35] can be seen as a kind of extended loss function. Here, two networks, a generator and a discriminator compete against each other. The generator tries to generate realistic data while the discriminator distinguishes between the generated data and the data from the training database. After successful training, the discriminator is not needed any longer, its mere purpose lies in providing a better loss for the generator. The reason why adversarial training is interesting for training generative models like BBWEs is due to their ability to model some modes of a distribution without smoothing or averaging over all modes.

B. Class of Networks

Another important aspect in the design of a DNN is the choice of class of networks to be used. Of popular choice are fully connected layers [18], [21], convolutional neural networks (CNN) [11], [36] or recurrent neural networks (RNN), with their known sub-types called long short-term memory (LSTM) units [37], [38], [8] or gated recurrent units (GRU) [39], [38], [34]. Fully connected layers are only used in systems that operate on frames [18], [21] while RNNs and CNNs allow for processing of time-domain data in a streaming way [23], [24], [34].

C. Autoregressive Networks

A remarkable contribution to field of generative DNN models was WaveNet [32], a model first used for speech synthesis. In this work and their previously released PixelCNN [40], the authors introduced several innovations. WaveNet models the

speech distribution as a product of conditional probabilities and a compact feature representation $\boldsymbol{h}$:

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}, \boldsymbol{h}), \quad (2)$$

where x_t is a speech sample at time t. Each audio sample is therefore conditioned on previous samples. This is implemented with causal convolutions. As a result, the network predicts samples that are fed back into the network. This is different to RNNs, in which the network architecture is autoregressive, whilst the training does not depend on generated samples. Furthermore, they use dilated convolution with gated activation units and conditioning:

$$\boldsymbol{z} = \tanh(K_{f,k} * x) \odot \sigma(K_{g,k} * x)), \quad (3)$$

in which $*$ denotes a convolution operator, $\odot$ denotes an element-wise multiplication operator, $\sigma()$ denotes a sigmoid function, k is the layer index, f and g denote filter and gate, respectively, and K is a learnable convolution filter kernel.

WaveNet has also been adopted for BBWE. In [41], it is trained on clean speech, conditioned with bitstream parameters of coded NB speech. Here the network acts as a decoder, implicitly doing bandwidth extension. Following this, in [24] WaveNet is conditioned with features calculated on NB signal. After successful training, only the features are fed to the network and the NB speech signal is neglected.

While WaveNet-based models claim very high perceptual quality, they are hard to train and the computational complexity at evaluation time is very high. This gave rise to several optimisations and alternative models (e.g. [42]). One particular alternative is LPCNet, originally designed for either speech synthesis [33] or speech coding [43]. In LPCNet the convolutional layers of WaveNet are replaced by recurrent layers. Furthermore the speech signal is decomposed into excitation signal and an envelope - similar to speech codecs [2], [4]. This is accomplished with linear predictive coding (LPC). The recurrent layers merely model the excitation signal, which is easier to predict. LPCNet has also been adopted for BBWE [34].

III. Proposed Systems

We propose three BBWEs based on DNNs, two based on convolutional architectures, the other one based on a mixture of convolutional and recurrent architectures. All are trained adversarial with the same discriminator, the same perceptual loss and the same optimisation algorithm. The architecture of the first BBWE is inspired by WaveNet, the other architectures are inspired by LPCNet. First, all generator networks are presented and since all systems share the same discriminator, it will be described at the end of this section.

A. Convolutional BBWE

The first architectural proposal for this task is a stack of convolutional neural networks (CNNs) as this is currently the standard building block of GANs. Using CNNs enables fast processing especially on GPUs.

We adopted a WaveNet-like structure for the convolutional generator model. Specifically, it is a stack of 20 layers where each layer uses causal convolutions with a kernel size of 33 and softmax-gated activations [44] for all layers. Biases have been omitted. One of these layers is displayed in Fig. 1. Each of the CNN layers has 32 input channels and 64 output channels. Half of the output channels are fed into $tanh$-activations and the other half is fed into softmax activation. Both activations are multiplied over the channel dimension in order to form the 32 channel output of each layer. This type of activation is more robust against reconstruction artefacts than both ReLU and sigmoid-gated activation.

An additional input layer maps the one-dimensional input signal to a 32-dimensional signal and an additional output layer maps the 32-dimensional signal back to a one-dimensional output signal.

The weights of the convolutional kernels are normalised using weight normalisation [45] to enable stable training behaviour. We also apply batch normalisation to the output features from the CNN layers to speed up the training process. Accordingly, a complete convolutional layer consists of causal convolution followed by batch normalisation and finally the softmax-gated activation to obtain the final output. There is also a residual connection or shortcut from the input to the output in order to avoid vanishing gradients and maintain stable and effective training [46].

In this convolutional BBWE, the model runs on raw speech waveforms in time domain. The input signal is firstly resampled from NB to WB using a simple Sinc interpolation, then it is fed to the generator model. The generator takes care of extending the original bandwidth of this upsampled signal reliably to get a complete WB structure with clearly higher perceptual quality.

This system is called CNN-GAN.

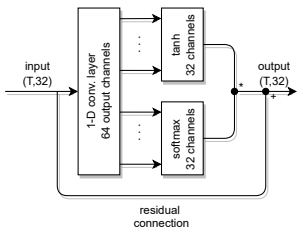

Fig. 1. Single layer of the CNN-GAN with softmax-gated activations. The CNN layer has 1-dimensional kernels with 32 input channels and 64 output channels. Half of the output channels are fed into $tanh$-activations and the other half is fed into softmax activation. The residual connection avoids vanishing gradients and maintains a stable and effective training.

B. LPC-GAN

This section proposes two systems that differ from the convolutional one in two aspects: First the architecture of the DNNs differs, second, the speech signal is decomposed into an excitation signal and an envelope. This is inspired by the BBWE based on LPCNet [34], but with fundamental structural differences. The motivation for decomposing the signal into excitation and envelope is the same as for the BBWE based

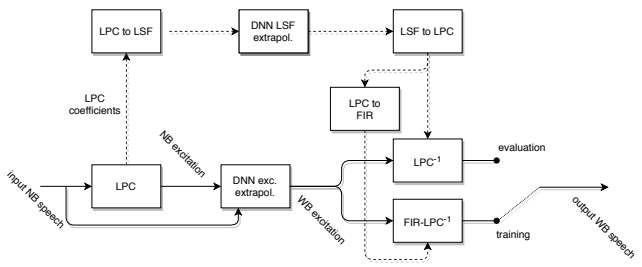

Fig. 2. Proposed system based on the decomposition of the speech signal into excitation signal and LPC envelope. All paths with solid lines operate in samples, all paths with dashed lines operate on frames of 15 ms.

on LPCnet [34], namely the reduction of computational complexity of the whole systems. Fig. 2 shows a block diagram of the system and Fig. 4 shows one of the DNNs bandwidth extending the excitation signal in detail.

Both systems only differ in the DNNs used for extrapolating the excitation signal. Fig. 2 shows a block diagram of the common processing and the DNNs bandwidth-extending the excitation signal are shown in detail in separate Fig. 4 and 1. Since the first architecture is based on recurrent layers, it is called LPC-RNN-GAN. The second architecture uses the same convolutional layers as the CNN-GAN and is called LPC-CNN-GAN.

In Fig. 2, the input NB speech signal is separated into LPCs representing the spectral envelope and an excitation signal (a.k.a residual). The excitation signal and the input signal are fed to the first DNN for extrapolation to a WB excitation signal. This path operates on samples, shown here as solid lines. The LPCs are extrapolated to a WB envelope with a second DNN in the upper path. This path operates on frames of 15 ms, shown here as dashed lines. Since LPC coefficients are IIR filter coefficients and manipulations like extrapolation could result in an unstable filter, they are extrapolated in the LSF domain [47]. LSFs are a bijective transformation of LPCs with several advantages: First, they are less sensitive to noise disturbances and an ordered set of LSFs with a minimum distance between the coefficients will always guarantee a stable LPC filter. Second, the spectral envelope at a particular frequency depends mostly on one of the LSFs so an erroneous extrapolation of a single LSF coefficient mainly affects the spectral envelope at a limited frequency range. These properties make them suitable for being extrapolated to a set representing a WB envelope. The extrapolated LSF coefficients are transformed back to the LPC domain for shaping the extrapolated excitation signal, which forms the output signal. This is achieved in different ways for training and evaluation.

The extrapolated excitation signal, shaped by the LPC envelope, forms the output WB signal. While training the DNN that extrapolates the excitation signal, the gradient needs to be propagated through the LPC filter, which can be achieved when the LPC filtering is performed by an additional DNN layer. Since the LPC filter is a pure IIR filter, this DNN layer should be a layer with recurrent units. Unfortunately, backpropagating gradients through a recurrent layer will cause the gradient to vanish (a.k.a. vanishing gradient problem [37]) and result in poor training. As a solution to this problem, the IIR filter coefficients are transformed into FIR filter coefficients by calculating the truncated impulse response from the IIR filter. It is known from signal processing that any IIR filter can be approximated by an FIR filter by truncating the infinite impulse response [48]. Then, the LPC shaping can be implemented with a convolutional layer. Fig. 3 shows the effect of truncating it to 64 samples. While the IIR LPC envelope is smooth, the truncated FIR envelope has lots of ripples and does not follow well the IIR envelope in high frequencies. For this reason the LPC coefficients are multiplied with an exponential function before calculating the truncated impulse response:

$$\hat{a}_i = a_i \cdot 0.8^i \quad \text{for } i = 0, \ldots, 12 \tag{4}$$

Here a_i are the IIR LPC coefficients calculated by the Levinson-recursion. The resulting $\hat{a}_i$ coefficients have less pronounced poles and are suitable for calculating the FIR envelope as shown in Fig. 3. However, less pronounced poles result in less shaping and thus not being as efficient as pure IIR coefficients.

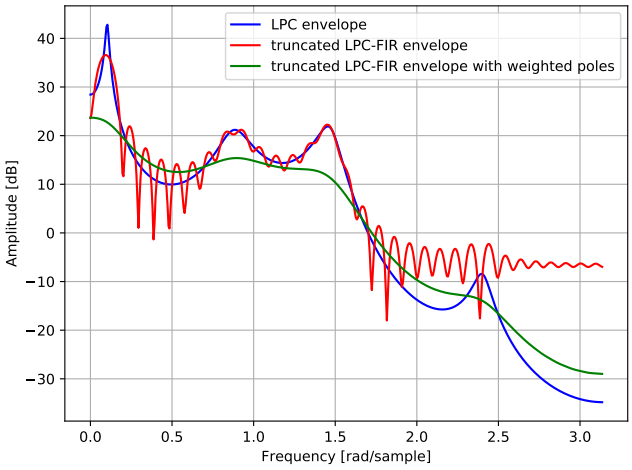

Fig. 3. Transfer functions of an IIR LPC filter of order 12 and FIR filters resulting from a truncated impulse response. The impulse response was truncated to 64 samples. For the filter shown in green, the IIR LPC coefficients were processed with Eq. 4, for the filter shown in red no processing was used.

Initial experiments have shown that the FIR shaped signal contains artefacts, which could easily be identified by the discriminator. As a result, the adversarial loss was not balanced and the generator was training poor. This could be solved by calculating the adversarial loss on the real and generated *unshaped* excitation signal.

The LPC shaping by an FIR filter is done only during training time. During evaluation time, no gradient needs to be backpropagated, so the LPC coefficients are applied as an IIR filter.

As already mentioned, two different DNNs are used for extrapolating the excitation signal, the first (LPC-RNN-GAN) is based on a mixture of convolutional and recurrent layers, the second (LPC-CNN-GAN) on convolutional architectures only. The first is shown in detail in Fig. 4. At first are 4 convolutional

layers followed by two recurrent layers with GRUs [39]. Since we want to compare the performance with the BBWE based on LPCnet [34], the GRUs have the same size as the GRUs in LPCnet. Their matrices are of size 256 x 256 and 256 x 16 respectively. A GRU layer computes for each time index t in the input sequence the following operation:

$$r_t = \sigma\left(\boldsymbol{W}_{ir} x_t + b_{ir} \boldsymbol{W}_{hr} h_{t-1} + b_{hr}\right) \quad (5)$$

$$z_t = \sigma\left(\boldsymbol{W}_{iz} x_t + b_{iz} \boldsymbol{W}_{hz} h_{t-1} + b_{hz}\right) \quad (6)$$

$$n_t = \tanh\left(\boldsymbol{W}_{in} x_t + b_{in} r_t \odot \left(\boldsymbol{W}_{hn} h_{t-1} + b_{hn}\right)\right) \quad (7)$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1} \quad (8)$$

$$(9)$$

where h_t being the hidden state at time t, $x_t$is the input at time t, h_{t-1} is the hidden state at time $t-1$, and r_t, z_t, n_t are the reset, update, and new gates, respectively. σ is the sigmoid activation function, and $\odot$ is the Hadamard product.

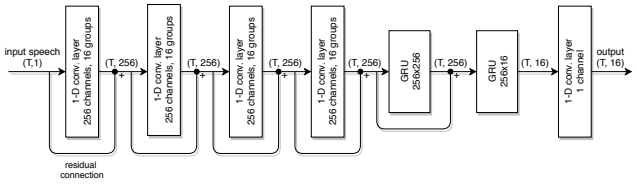

Fig. 4. Structure of the DNN extrapolating the excitation signal. Shapes of the signals given in brackets omitting the batch dimension. T is the length of the input signal.

The purpose of the initial CNN layers is to add a feature dimension to the one-dimensional time-domain signal. This feature dimension is needed by the GRU layers, otherwise the matrices in the GRUs would collapse to simple vectors. CNNs add the feature dimension by operating kernels in parallel, usually phrased as channels. Consequently 256 channels are needed so that the CNN layers and GRU layers are compatible. This would result in high computational complexity, which can be prevented by splitting the channels into 16 groups of each 16 channels. This is the same as having 16 layers of each 16 channels in parallel. The structure of the CNN layers (kernel size, gated activation etc.) is the same as described in Sec. III-A. Since the output of the second GRU layer still has a feature dimension, it is squeezed to a one-dimensional signal with a single convolutional kernel with kernel size 1.

The main contribution of computational complexity comes from the matrices in the first GRU. To reduce the complexity further, these matrices can be made sparse during training [49]. After initial training iterations with dense matrices, blocks with low magnitude are identified and forced to zero. A boolean matrix stores the indices of those blocks. With proceeding training, more blocks are forced to zero, until a desired sparseness is achieved. Similar to [33] 16x1 blocks are used while also including all diagonal terms. The final percent of elements preserved of the matrices are:

$\boldsymbol{W}_{ir}, \boldsymbol{W}_{hr}$	5 %
$\boldsymbol{W}_{iz}, \boldsymbol{W}_{hz}$	5 %
$\boldsymbol{W}_{in}, \boldsymbol{W}_{hn}$	20 %

Neglecting the computational overhead of indexing, this sparsification scheme reduces the computational complexity of the GRU by 90%. Fig. 5 shows one of the sparse matrices after training.

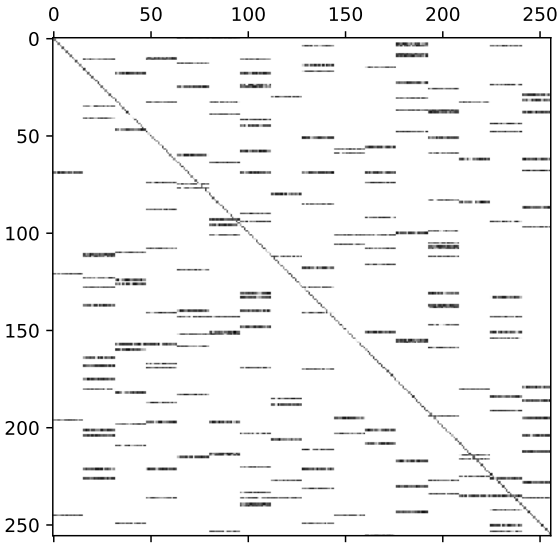

Fig. 5. One of the matrices from a GRU after sparsification.

The DNN only based on convolutional architectures has the same structure as the one described in Sc. III-A with three structural differences. First, the size of the CNN kernels is only 17 and second, to compensate the resulting smaller receptive field, this system uses dilated convolutions with a dilation factor of 2 per layer. Third, to save complexity, this system makes use of the above mentioned grouping, by splitting the channel-dimension into 4 groups. In Sec. V-A2 it will be shown that by this the computational complexity can be reduced by a factor of about 3.

The DNN extrapolating the LPC envelope is also a combination of CNN layers followed by a GRU layer and a final CNN layer. The CNN layers have two-dimensional kernels with kernel size 3 and they operate on the current, one past and one future frame and are the main source of algorithmic delay of the whole system.

C. Discriminator

The discriminator acts as a convolutional encoder that extracts a latent representation of the input signal to evaluate the adversarial loss. The CNN-GAN, LPC-CNN-GAN and LPC-RNN-GAN use the same discriminator architecture for the adversarial training, consisting of convolutional layers. A stable adversarial training is achieved by applying spectral normalisation to the convolution kernels of the discriminator layers [50]. This kind of normalisation enforces the Lipschitz condition to the function learned by the discriminator, which was found important for an effective and stable adversarial training procedure. The discriminator operates in conditional setting [51], hence the input signal includes the real/fake WB speech waveform concatenated with the upsampled NB one along the channel dimension. Fig. 6 depicts the discriminator.

It consists of 6 convolutional layers, with kernel size of 32 and stride of 2 steps. Biases have been omitted. For activation, we use Leaky ReLU with negative slope of 0.2.

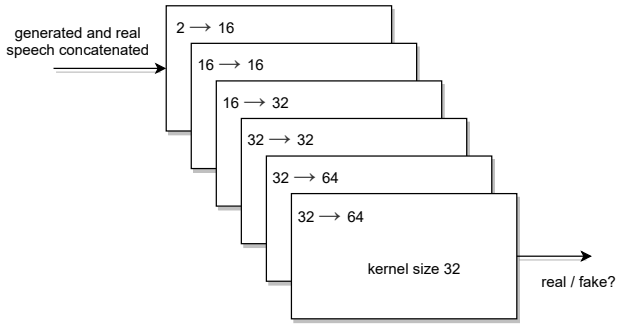

Fig. 6. The GAN discriminator network consisting of 6 convolutional layers with each layer having kernels of 32 samples operating at strides of 2. The numbers in the layers represent the input and output channel dimension of each layer.

Since the conditioning input is time domain NB speech, the discriminator will reject generated speech with a different waveform as the original waveform. The LPCNet based BBWE imposes less constraints on the generated waveform as explained later in Sec. V-D. In order to have a GAN that imposes less restrictive constraints to the generated waveform, an second discriminator is evaluated that gets a low dimensional feature representation as input. The features are Mel-frequency cepstral coefficients (MFCCs) [52] calculated on the NB speech. This discriminator, together with the absence of any L^p-loss, tends to penalise less generated speech with a waveform different to the original.

D. Training Objective

The adversarial metric used in this work is Hinge loss [53]:

$$L_{hinge} = max(0, 1 - D()), \tag{10}$$

where $D()$ is the the raw output of the discriminator. Lim et. al. [53] showed that Hinge loss has less mode collapse and a more stable training behaviour compared to the loss used in the initial GAN paper [35] or the Wasserstein distance [54]. Initial experiments with the proposed systems have shown that hinge loss performs similar as feature matching.

As already observed in [30], [25] the adversarial loss can be amended by an L^p-norm calculated on samples and on features. Here we use the L^1-norm calculated on time domain samples and as feature loss L_{mel} the L^2-norm calculated on logarithmic Mel energies. The total loss training the generator is:

$$L = (1 - \lambda)L_{hinge} + \lambda(L^1 + L_{mel}) \tag{11}$$

IV. Experimental Setup

As training material we used several publicly available speech databases [55], [56], [57] as well as other speech items of different languages. In total, 13 hours of training material were used, all of it resampled to 16 kHz sampling frequency. Silent passages in the training data were removed with a voice-activation-detection [58]. The NB input signal was coded with AMR-NB at 10.2 kbps. The target clean speech signal was pre-emphasised with a first order filter E

$$E(z) = 1 - 0.68z^{-1}. \tag{12}$$

The inverse (de-empahsis) filter D

$$D(z) = \frac{1}{1 - 0.68z^{-1}} \tag{13}$$

was applied to the generated speech. The reason for this is to compensate the spectral tilt of speech which could result in less pronounced high frequencies in the generated speech. The LPC envelope of order 12 is extracted on frames of 128 samples windowed with a Hann window by calculating the time-domain autocorrelation followed by the Levinson recursion. Thereafter they are converted to an FIR filter as explained in Sec. III-B. The DNNs are trained with batches of 8 items with each items containing 1 second of speech.

The optimisation algorithm for both, the generator and discriminator is Adam [59] with a generator learning rate of 0.0001 and a discriminator learning rate of 0.0004. For a more stable adversarial loss, the coefficients used for computing running averages of the gradient and its square (the beta-parameters) are set to 0.5 and 0.99 respectively. Since RNNs of the LPC-RNN-GAN (see Sec. Sec. III-B) usually train slower than CNNs, the learning rate is set to 0.0001 for generator and discriminator. The beta-parameters for training the generator are set to 0.7 and 0.99. The factor λ controlling the amount of adversarial loss in Eq. 11 is set to 0.0015. The Sparsification of the GRU layer starts at the 160th batch and the final sparseness is achieved at the 10000th batch.

All CNN layers have been trained with batch normalisation for faster training and to prevent the networks falling into mode collapse.

The additional frame-rate network extrapolating LPC coefficients in the LSF domain has 10 CNN layers followed by a single GRU and a final CNN layer. The initial CNN layers are two-dimensional convolutions with kernel size 3x3, 16 channels, $tanh$-activation functions and residual connections. The GRU has a matrix size of 16x16 and the final convolutional layer with 5 channels, the number of missing LSF coefficients concatenated to the NB LSF coefficients to form the WB LSF coefficients.

In Sec. V, the presented systems are compared to an LCPNet based BBWE published in [34]. In contrast to the published system, the DNN used for extrapolating the LPC envelope has here been trained adversarial. For this, the same discriminator architecture has been used, with only adapting the input dimension.

All DNNs were implemented and trained with PyTorch [60].

V. Evaluation

The perceptual quality of the presented BBWEs is evaluated by objective measures previously used to access the quality of speech and subjectively by a listening test. Furthermore, the algorithmic delay and computational complexity are given for each BBWE. Correlation between objective and subjective results are studied to see if they are powerful enough to predict the subjective assessment.

A. Computational Complexity

The computational complexity of the proposed BBWEs is an estimate of weighted million operations per second (WMOPS) per speech-sample. WMOPS is the ITU unit for calculating computational complexity [61] of standardised speech processing tools. Additions (ADD), multiplications (MUL) as well as multiply-add (MAC) operations are each counted as one operation while complex operations like $tanh$, sigmoid or softmax operations each count as 25 operations. In the following sections, the number are calculated per speech-sample. This number is multiplied by the sampling frequency to get an estimate of the WMOPS. This should be seen as a rough approximation that doesn't consider advantages of todays parallel processing architectures. The results are summarized in Tab. I together with the computational complexity of EVS [2], [62], the state-of-the-art standardised speech codec.

1) Computational Complexity of CNN-GAN: The complexity of one convolution of one of the kernels of the CNN layer depends only on the kernel size that is denominated here as K. This needs K MAC operations. A CNN layer with N_i input channels and N_o output channels has $N_i * N_o$ convolutional kernels and, like in fully connected layers, all possible channel combinations are executed. As a result $N_i * N_o * K$ MAC operations are executed. As mentioned in Sec. III-A the output of the CNN layer is split into two parts, one going into a $tanh$-activation function, the other one going into a softmax activation function followed by a element-wise product. Furthermore the calculation of the residual connection needs N_i ADD operations. Since the number of output channels is $N_o = 2 * N_i$ due to the gating mechanism, one convolutional layer executes $N_o^2 * 2 * K + 2 * N_o * 25 + N_o * 2$ operations. The initial and final convolutional layer execute $N_o * K$ operations each. Tab. I summarises the number of operations for $N_o = 64$ channels, kernel size $K = 32$ and 22 layers in total.

2) Computational Complexity of LPC-RNN-GAN and LPC-CNN-GAN: As mentioned on Sec. III-B, this system has initial CNN layers that split the one-dimensional signal into 256 channels. These layers are the same CNN layers as above with the difference that the channels are grouped to blocks described in Sec. III-B. Here a total of 256 channels are grouped to 16 blocks of each 16 channels. This is the same as having 16 CNN layers with 16 channels in parallel.

The operations of one RNN layer for a single speech-sample are given in Eq. 5. Lets denominate M_i as the input dimension and M_h as the output (or hidden) dimension. Then the calculations of the reset and update gates (first two lines in the equation) each need $M_i * M_h * 2$ MAC operations plus M_h sigmoid operations. The *new* gate (third line of the equation) needs $M_i * M_h * 2 + M_h$ MAC operations plus M_h tangents hyperbolicus operations. Finally the output (last line) needs $M_h * 2$ MAC operations. Since the first, large GRU layer uses sparsified matrices (see Sec. III-B), the operations are calculated for the reduced matrix sizes. Overhead due to additional addressing-operations are neglected. For the first GRU all matrices are square with $M_i = M_h = 256$, for the second GRU $M_i = 256$ and $M_h = 32$.

The final CNN layer just adds up the output dimension and needs 32 ADD operations.

The computational complexity of the LPC-CNN-GAN is calculated as in Sec. V-A1 as having 4 such networks with a channel dimension of only 8 in parallel.

At evaluation time, the LPC filter is applied as IIR filter with 12 taps and only needs 12 MAC operations per sample. The conversion of LPC to LSF coefficients and back will be disregarded here, since these conversions are done on a frame base and their contribution to the overall complexity is expected to be small. Tab. I summarises the number of operations with the used parameterisation.

B. Algorithmic Delay

The algorithmic delay is the theoretical delay in ms between the input speech and the processed output speech caused by block-processing of speech samples. CPU or GPU time is not considered. The numbers are summarises in Tab. I.

1) CNN-GAN: The source of algorithmic delay of the CNN-GAN are the convolutional operations with kernels of size K. Each convolutional layer adds an algorithmic delay of $\lfloor K/2 \rfloor$ samples, since $\lfloor K/2 \rfloor - 1$ tabs of the kernel are calculated on previous samples and do not contribute to the delay. The overall system with 22 convolutional layers and kernels of size 33 has a total delay of 353 samples or 22.0 ms at 16 kHz sampling frequency.

2) LPC-RNN-GAN and LPC-CNN-GAN: The source of algorithmic delay of these systems are the initial convolutional layers and the LPC processing. The GRU layers do not introduce any algorithmic delay. The 4 convolutional layers have a kernel size of 16 tabs, with 8 tabs calculated on future samples, hence a delay of 2 ms. Thus the algorithmic delay of the LPC processing, resulting from a windowed autocorrelation function is 15 ms. Since this block processing is independent form the convolutional layer, the total algorithmic delay of the whole system is 15 ms. The LPC-CNN-GAN uses kernels with half the size as the CNN-GAN but with dilation of 2 and thus has the same algorithmic delay as the CNN-GAN.

TABLE I
COMPUTATIONAL COMPLEXITY AND ALGORITHMIC DELAY OF THE PROPOSED SYSTEMS, THE PREVIOUSLY PUBLISHED LPCNET-BBWE [34] AND EVS [2], [62], A STATE-OF-THE-ART STANDARDISED SPEECH CODEC. WMOPS IS THE ITU STANDARD FOR CALCULATING COMPUTATIONAL COMPLEXITY [61] AND CALCULATED AT A SAMPLING FREQUENCY OF 16KHZ.

	OPS per sample	WMOPs	algorithmic delay
CNN-GAN:	1387897	22206	22 ms
LPC-RNN-GAN:	649286	10388	15 ms
LPC-CNN-GAN:	383353	6133	22 ms
LPCNet:	130092	2081	15 ms
EVS:	-	88	32 ms

C. Objective Perceptual Quality

While a listening test with human listeners is the ultimate base for evaluating the perceptual quality, it takes quite some effort to conduct. Objective measures are an easy to use alternative. Here four different measures have been used,

Perceptual Objective Listening Quality Analysis, Fréchet Deep Speech Distance, Word Error Rate and Short-Time Objective Intelligibility measure. All measures except the Word Error Rate are calculated on a multilingual, multiple speaker database of about 1 hour not being part of the training set.

1) Perceptual Objective Listening Quality Analysis: Perceptual Objective Listening Quality Analysis (POLQA) is a standardised method that aims to predict the perceptual quality of coded speech signals on the same Mean Opinion Scale (MOS) used in listening tests [63]. The estimated results are summarized in Fig. 7 and show that the LPC-RNN-GAN achieves the highest ratings, followed by the CNN-GAN.

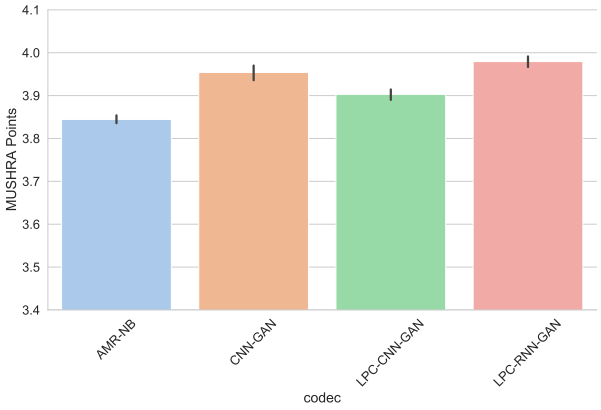

Fig. 7. Perceptual Objective Listening Quality Analysis (POLQA) of different BBWEs with 95% confidence intervals. Higher values mean better quality.

2) Fréchet Deep Speech Distance (FDSD): The evaluation of the quality of speech or images generated by GANs is a difficult task. In the typical use case GANs generate items from noise and metrics based on an L^p-norm can not be used since there is no reference to compare with. A common objective measure to assess the quality of images created by GANs is the Fréchet inception distance (FID) [64]. This metric is calculated on the output of a different DNN trained to classify images or speech. Opposed to generative modelling, image and speech classification (recognition) is already quite elaborated and the entropy of the output of a DNN classifying the generated data might give an estimate of the quality. Items that are classified strongly as one class over all other classes indicate a high quality and the conditional probability of generated items should have a low entropy. Furthermore, GANs should generate a large variety of items (not suffer from mode collapse) and therefore the integral of the marginal probability distribution of the classification output is preferred to have a high entropy. The inception distance (ID) in [65] formulates this mathematically. Heusel et. al. [65] have improved this by also using the distribution of classification results of real data based on the Fréchet distance:

$$\text{FID} = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}). \quad (14)$$

Here $\mu_r, \mu_g, \Sigma_r, \Sigma_g$ are the mean and covariance of the output of a classification network of real and generated data respectively. The Fréchet Deep Speech Distance (FDSD) proposed by Binkowski et. al. [66] uses the *DeepSpeech 2* speech recognition network [67] to calculate the Fréchet distance that is also used in this work. Fig. 8 gives the FDSD scores of the different BBWEs.

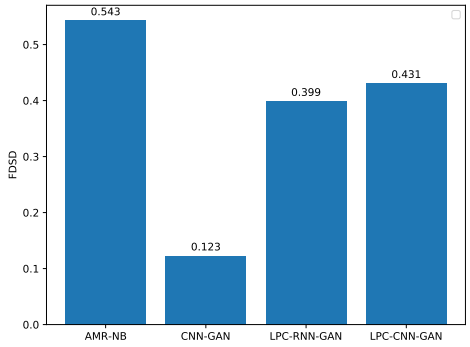

Fig. 8. Fréchet Deep Speech Distance (FDSD) of different BBWEs. Lower values mean better quality.

3) Word Error Rate: Besides improving the perceptual quality, a BBWE can also improve the intelligibility of speech [5], [6] and furthermore, the performance of Automatic Speech Recognition (ASR) systems. State of the art ASR systems are based on DNNs trained on speech with a fixed sampling frequency, mostly 16 kHz. As a result the performance of such systems drops significantly when the speech is coded with a NB codec. This section evaluates the impact of coding speech with AMR-NB on the word error rate (WER) of a state of the art ASR system and how BBWE can mitigate this impact. The ASR system used here is Mozillas open implementation of the RNN based *DeepSpeech* system [68] with Connectionist temporal classification (CTC) loss [69] trained on the common voice multilingual speech corpus [70]. The evaluation is done on the evaluation set from this database. The WER metric is evaluated at the word level of the transcribed speech and computes:

$$WER = \frac{S + D + I}{S + D + C} \quad (15)$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and C is the number of correct words of a transcription. Fig. 9 depicts the ASR performance of AMR-NB and the different BBWEs together with the character error rate (CER) which is calculated similar to WER but on a character level instead of a word level.

Tab. II gives an example of one of the worst performing items. It is interesting to see, that although uncoded items perform better in average, there are no outliers with a performance worse than 0.6 with AMR-NB coded items from the database. BBWE processed items improve the average WER but also produce outliers with a WER of 8.0 and more.

4) The Short-Time Objective Intelligibility measure (STOI): The Short-Time Objective Intelligibility measure (STOI) is defined as an estimate of the linear correlation coefficient between the temporal energy envelopes of clean and BBWE-processed speech sub-bands. These sub-bands are calculated on a time-frequency-representation, obtained from segmenting speech signals into 50% overlapping, Hann-windowed frames with a length of 256 samples, where each frame is zero-padded

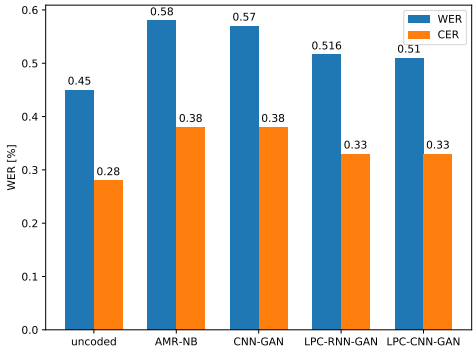

Fig. 9. Word error rate (WER) and character error rate (CER) of different BBWEs. Lower values mean better performance.

TABLE II
EXAMPLES OF WORST CASE ASR PERFORMING ITEMS.

	WER	original	result
uncoded	1.67	"i'm not drivel-ling"	"i am at the level"
AMR-NB	0.6	"every purchase is a vote"	"are purchases a vote"
CNN-GAN	9.0	"undefined"	"the thing honour and he bent on the corner"
LPC-RNN-GAN	9.0	"undefined"	"the thing honour and even on the corner of"
LPC-CNN-GAN	8.0	"undefined"	"everything over and he banished round the corner"

up to 512 samples and Fourier transformed. 15 one-third octave bands are calculated by averaging DFT-bins. Originally this measure is calculated on speech sampled with 10kHz sampling frequency. Since we are assessing the quality of WB speech, this measure is extended to 16 kHz. Fig. 10 shows the results for the presented systems. According to this measure the LPC-RNN-GAN performs best, followed by the LPC-CNN-GAN.

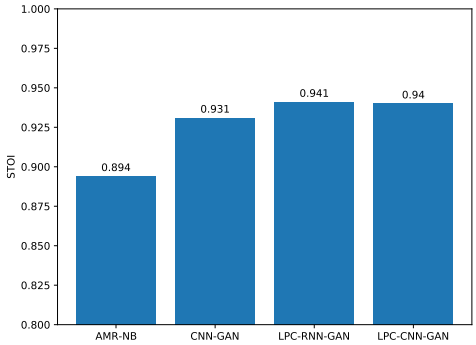

Fig. 10. Short-Time Objective Intelligibility measure (STOI) for the presented systems. Smaller values mean lower quality.

D. Subjective Perceptual Quality

To ultimately judge the perceptual quality of the proposed systems, a MUSHRA listening test [71] was conducted. According to the MUSHRA methodology, the test items contain the reference marked as such, a hidden reference and the AMR-NB coded signal serving as anchor. 12 experienced listeners participated in the test. The speech items used in the test are about 10 seconds long and neither part of the training nor the test set. The items contain Chinese, English, French, German and Spanish speech from native speakers. The results are presented in Fig. 11 per item and in Fig. 12 averaged over all items as box plots with mean values and 95% confidence intervals. Fig. 13 shows the results as bar plot.

The system marked as *CNN-feat-cond* is the CNN-GAN trained with a discriminator whose conditional input is based on features as explained in Sec. III-C. The L^1-loss is also removed form the training objective.

The results show that all presented systems significantly improve the quality of AMR-NB speech for all items. Except for the CNN-feat-cond, none of the presented systems is significantly better than the others. The tendentially best system is the LPC-CNN-GAN which is also significantly better than the CNN-feat-cond system.

Inspecting the results from single items it strikes that the quality is fairly dependent on the item. The LPC-CNN-GAN is not always the best performing system. For Spanish female, German female and male 2 items, the LPCNet based system performs best. For the Chinese male items the LPC-RNN-GAN performs best, for the Spanish male item the CNN-GAN performs best. The CNN-GAN often has the fewest noisy artefacts but frequently fails to reconstruct fricatives well.

The variance in quality is especially high for the LPCNet based system. This system sometimes delivers very high quality with occasional severe artefacts like clicks and unstable pitch. The GAN based systems, on the other hand, do not suffer from such severe artefacts but from broadband crackling noise. The LPCNet based system, and also sometimes the feature conditioning based system, change the characteristic of the voice since both systems impose less constraints on the generated waveform. In a MUSHRA test this can result in lower scoring as in different test methods like Absolute Category Rating (ACR) tests where the reference is not given.

In order to see how well the objective measures reflect the subjective assessments, the correlation with MOS values from the listening test is studied. For fair comparison all measures are normalised to zero mean and standard deviation. Since FDSD, WER and CER are giving lower values for better quality estimates, their values are negated first. Fig. 14 shows the normalised values and Tab. III the correlation values.

It can be seen that STOI has the highest correlation with the MOS value, followed by POLQA, WER and CER.WER, however, is the only measure that has the same order as the listening test results. The difference between WER and FDSD values is strange, since both measures are based on the output of similar networks (*DeepSpeech* and *DeepSpeech 2*).

VI. CONCLUSION

This work presents novel approaches for BBWE based on generative models used for bandwidth extension of speech signals. For two of the presented systems, an established paradigm from the speech coding world, namely the decomposition of the speech signal into envelope and excitation signal

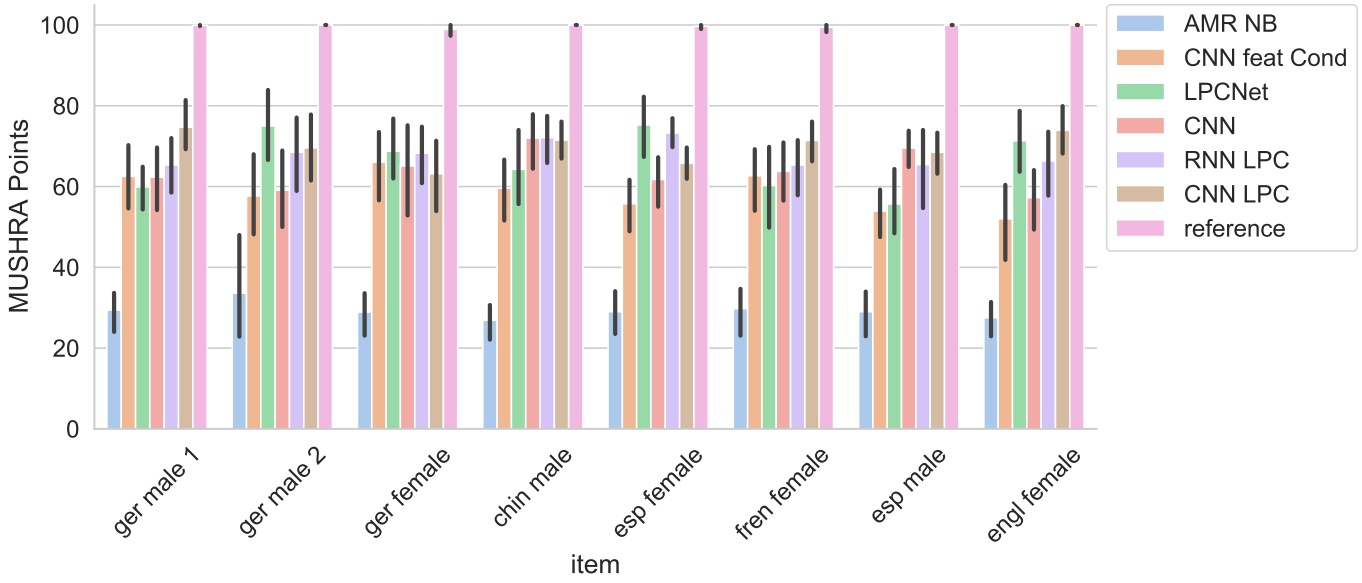

Fig. 11. Results from listening test evaluating different BBWEs as boxplot with 95% confidence intervals per item.

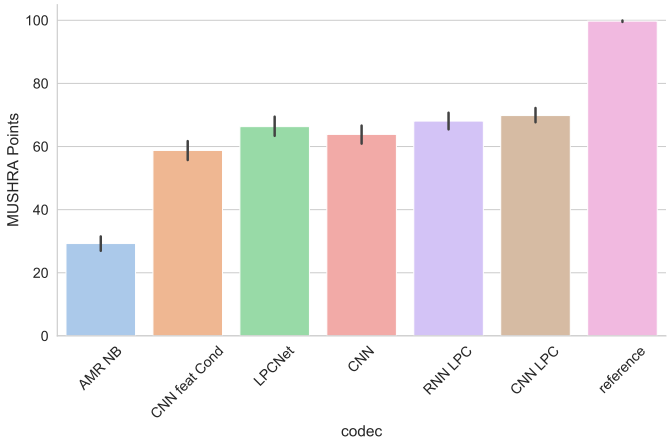

Fig. 12. Results from listening test evaluating different BBWEs as bar plot with 95% confidence intervals averaged over all items.

TABLE III
CORRELATION OF SUBJECTIVE MOS VALUES WITH OBJECTIVES MEASURES

STOI:	0.99
FDSD:	0.518
WER:	0.76
CER:	0.68
POLQA:	0.79

known as the source-filter model, has been applied to GAN models. As a result, the computational complexity can be lowered by a factor of about 3. This approach was tested and evaluated within the application of BBWE but is not limited to it. Furthermore, this system improves the speech recognition error rate of NB speech by the highest margin.

Moreover, this work compares two fundamental different approaches to do BBWE, namely GAN models and an autoregressive model. Both approaches rely on generative models that are able to model complex data distributions, like the distribution of time domain speech and both approaches do not suffer from smoothing problems.

Both approaches have a moderate computational complexity compared to state-of-the-art models like WaveNet [32]. The LPCNet based BBWE is the model with the lowest computational complexity. The main reason for the lower complexity is, that this model imposes less constraints on the generated waveform. The waveform generated by LPCNet can be very different to the original waveform, while the GAN based BBWEs are preserving the original waveform due to conditioning and a mix of adversarial loss and L^1-loss. Unfortunately changing the conditioning to feature conditioning and removing the L^1-loss didn't improve the quality of the generated speech.

The LPC-RNN-GAN and the LPC-CNN-GAN differ in the DNN used for excitation signal extrapolation. The first is based on a mixture of CNNs and RNNs, the latter uses CNNs only. Both DNNs have about the same computational complexity. Although there is no significant difference in performance, the LPC-CNN-GAN performs tendentially better. In addition, the training time of CNNs is shorter and they are less delicate to hyperparameter tunings. The LPC-RNN-GAN successfully applies sparsification in the context of GAN training for the first time.

Correlating the results from the listening test with the objective measures gives ambiguous results. Although the authors in [66] showed that the FDSD measure is performing well in estimating the quality of adversarial generated speech, it fails here to access the small differences between the presented systems. The measure correlating best with the subjective results are the STOI and WER measures.

REFERENCES

[1] International Telecommunication Union, "Pulse code modulation (pcm) of voice frequencies," ITU-T Recommendation G.711, November 1988.

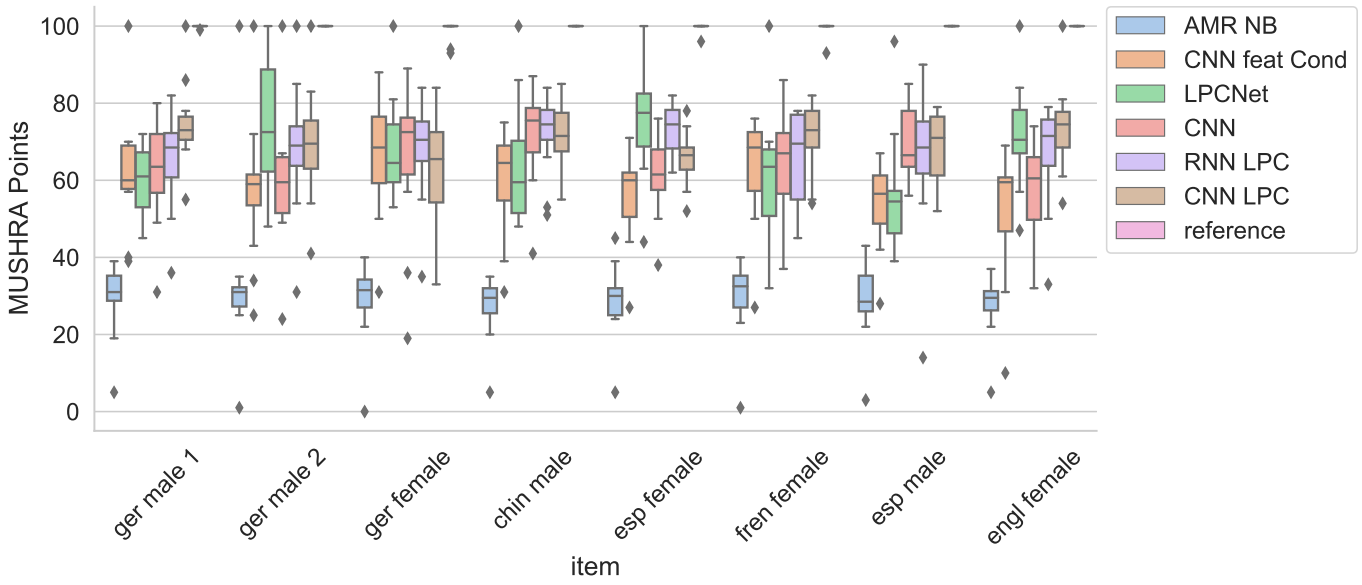

Fig. 13. Results from listening test evaluating different BBWEs as warm plot showing the ratings from each user.

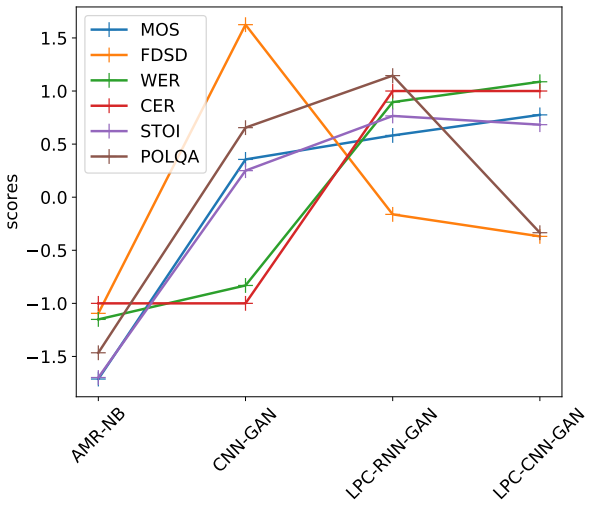

Fig. 14. Normalised objective and subjective measures

[2] S. Bruhn, H. Pobloth, M. Schnell, B. Grill, J. Gibbs, L. Miao, K. Järvinen, L. Laaksonen, N. Harada, N. Naka, S. Ragot, S. Proust, T. Sanda, I. Varga, C. Greer, M. Jelinek, M. Xie, and P. Usai, "Standardization of the new 3GPP EVS codec," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 5703–5707. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7179064

[3] S. Disch, A. Niedermeier, C. R. Helmrich, C. Neukam, K. Schmidt, R. Geiger, J. Lecomte, F. Ghido, F. Nagel, and B. Edler, "Intelligent gap filling in perceptual transform coding of audio," in *Audio Engineering Society Convention 141, Los Angeles*, Sep 2016. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=18465

[4] 3GPP, "TS 26.090, Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions," 1999.

[5] P. Bauer, R. Fischer, M. Bellanova, H. Puder, and T. Fingscheidt, "On improving telephone speech intelligibility for hearing impaired persons," in *Proceedings of the 10. ITG Conference on Speech Communication, Braunschweig, Germany, September 26-28, 2012*, 2012, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/6309632/

[6] P. Bauer, J. Jones, and T. Fingscheidt, "Impact of hearing impairment on fricative intelligibility for artificially bandwidth-extended telephone speech in noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 7039–7043. [Online]. Available: https://doi.org/10.1109/ICASSP.2013.6639027

[7] J. Abel, M. Kaniewska, C. Guillaume, W. Tirry, H. Pulakka, V. Myllylä, J. Sjoberg, P. Alku, I. Katsir, D. Malah, I. Cohen, M. A. T. Turan, E. Erzin, T. Schlien, P. Vary, A. H. Nour-Eldin, P. Kabal, and T. Fingscheidt, "A subjective listening test of six different artificial bandwidth extension approaches in english, chinese, german, and korean," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 2016, pp. 5915–5919. [Online]. Available: https://doi.org/10.1109/ICASSP.2016.7472812

[8] K. Schmidt and B. Edler, "Blind bandwidth extension based on convolutional and recurrent deep neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5444–5448.

[9] K. Schmidt, "Neubildung von unterdrückten sprachfrequenzen durch ein nichtlinear verzerrendes glied," Dissertation, Techn. Hochsch. Berlin, 1933.

[10] M. Schroeder, "Recent progress in speech coding at bell telephone laboratories," in *Proceedings of the third international congress on acoustics, Stuttgart*, 1959.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[13] X. Li, V. Chebiyyam, and K. Kirchhoff, "Speech audio super-resolution for speech recognition," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15-19, 2019*, 09 2019.

[14] P. Jax and P. Vary, "Wideband extension of telephone speech using a hidden markov model," in *2000 IEEE Workshop on Speech Coding. Proceedings.*, 2000, pp. 133–135.

[15] K. Schmidt and B. Edler, "Deep neural network based guided speech bandwidth extension," in *Audio Engineering Society Convention 147*, Oct 2019. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=20627

[16] H. Carl and U. Heute, "Bandwidth enhancement of narrow-band speech signals," in *Signal Processing VII: Theories and Applications: Proceed-*

ings of EUSIPCO-94 Seventh European Signal Processing Conference, September 1994, pp. 1178–1181.

[17] H. Pulakka and P. Alku, "Bandwidth extension of telephone speech using a neural network and a filter bank implementation for highband mel spectrum," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 7, pp. 2170–2183, 2011. [Online]. Available: https://doi.org/10.1109/TASL.2011.2118206

[18] K. Li and C. Lee, "A deep neural network approach to speech bandwidth expansion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 4395–4399. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178801

[19] P. Bauer, J. Abel, and T. Fingscheidt, "Hmm-based artificial bandwidth extension supported by neural networks," in *14th International Workshop on Acoustic Signal Enhancement, IWAENC 2014, Juan-les-Pins, France, September 8-11, 2014*, 2014, pp. 1–5. [Online]. Available: https://doi.org/10.1109/IWAENC.2014.6953304

[20] J. Sautter, F. Faubel, M. Buck, and G. Schmidt, "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 7005–7009.

[21] J. Abel, M. Strake, and T. Fingscheidt, "A simple cepstral domain dnn approach to artificial speech bandwidth extension," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5469–5473.

[22] J. Abel and T. Fingscheidt, "Artificial speech bandwidth extension using deep neural networks for wideband spectral envelope estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 71–83, 2018.

[23] Z. Ling, Y. Ai, Y. Gu, and L. Dai, "Waveform modeling and generation using hierarchical recurrent neural networks for speech bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 883–894, May 2018.

[24] A. Gupta, B. Shillingford, Y. M. Assael, and T. C. Walters, "Speech bandwidth extension with wavenet," *ArXiv*, vol. abs/1907.04927, 2019.

[25] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," 2019.

[26] Y. Dong, Y. Li, X. Li, S. Xu, D. Wang, Z. Zhang, and S. Xiong, "A time-frequency network with channel attention and non-local modules for artificial bandwidth extension," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6954–6958.

[27] J. Makhoul and M. Berouti, "High-frequency regeneration in speech coding systems," in *ICASSP '79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1979, pp. 428–431.

[28] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *CoRR*, vol. abs/1802.08435, 2018. [Online]. Available: http://arxiv.org/abs/1802.08435

[29] S. Li, S. Villette, P. Ramadas, and D. J. Sinder, "Speech bandwidth extension using generative adversarial networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5029–5033.

[30] S. E. Eskimez, K. Koishida, and Z. Duan, "Adversarial training for speech super-resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 347–358, 2019.

[31] X. Hao, C. Xu, N. Hou, L. Xie, E. S. Chng, and H. Li, "Time-domain neural network approach for speech bandwidth extension," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 866–870.

[32] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016, p. 125.

[33] J. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5891–5895.

[34] K. Schmidt and B. Edler, "Blind bandwidth extension of speech based on lpcnet," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2020, pp. 5444–5448.

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[36] Y. Gu and Z. Ling, "Waveform modeling using stacked dilated convolutional neural networks for speech bandwidth extension," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 1123–1127. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0336.html

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[38] Y. Gu, Z. Ling, and L. Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 297–301. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-678

[39] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS Deep Learning workshop, Montréal, Canada*, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[40] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," *CoRR*, vol. abs/1606.05328, 2016. [Online]. Available: http://arxiv.org/abs/1606.05328

[41] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "Wavenet based low rate speech coding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 676–680.

[42] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "Fftnet: A real-time speaker-dependent neural vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2251–2255.

[43] J.-M. Valin and J. Skoglund, "A real-time wideband neural vocoder at 1.6 kb/s using lpcnet," *ArXiv*, vol. abs/1903.12087, 2019.

[44] A. Mustafa, A. Biswas, C. Bergler, J. Schottenhamml, and A. Maier, "Analysis by Adversarial Synthesis - A Novel Approach for Speech Vocoding," in *Proc. Interspeech*, 2019, pp. 191–195. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1195

[45] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in NeurIPS*, 2016, pp. 901–909.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] Yao Tianren, Xiang Juanjuan, and Lu Wei, "The computation of line spectral frequency using the second chebyshev polynomials," in *6th International Conference on Signal Processing, 2002.*, vol. 1, Aug 2002, pp. 190–192 vol.1.

[48] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

[49] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," 2018.

[50] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018.

[51] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *ArXiv*, vol. abs/1411.1784, 2014.

[52] A. Salman, E. Muhammad, and K. Khurshid, "Speaker verification using boosted cepstral features with gaussian distributions," in *2007 IEEE International Multitopic Conference*, 2007, pp. 1–5.

[53] J. H. Lim and J. C. Ye, "Geometric gan," 2017.

[54] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

[55] C. Veaux, J. Yamagishi, and K. Macdonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[56] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 5206–5210. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964

[57] M. Sołoducha, A. Raake, F. Kettler, and P. Voigt, "Lombard speech database for german language," in *Proc. DAGA 2016 Aachen*, 03 2016.

[58] "Webrtc vad v2.0.10," https://webrtc.org.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani,

S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[61] ITU-T Study Group 12, *Software tools for speech and audio coding standardization*, Geneva, 2005.

[62] G. T. 26.445, "EVS codec; detailed algorithmic description; technical specification, release 12," Sep. 2014.

[63] ITU-T Study Group 12, *P.863 : Perceptual objective listening quality prediction*, Geneva, 2018.

[64] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017. [Online]. Available: http://arxiv.org/abs/1706.08500

[65] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf

[66] M. Binkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *CoRR*, vol. abs/1909.11646, 2019. [Online]. Available: http://arxiv.org/abs/1909.11646

[67] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. H. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[68] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. [Online]. Available: http://arxiv.org/abs/1412.5567

[69] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[70] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *CoRR*, vol. abs/1912.06670, 2019. [Online]. Available: http://arxiv.org/abs/1912.06670

[71] ITU-R, *Recommendation BS.1534-1 Method for subjective assessment of intermediate sound quality (MUSHRA)*, Geneva, 2003.