
# Time-domain Speech Enhancement Using Generative Adversarial Networks

Santiago Pascual, Joan Serrà, Antonio Bonafonte

PII: S0167-6393(19)30135-9
DOI: https://doi.org/10.1016/j.specom.2019.09.001
Reference: SPECOM 2664

To appear in: *Speech Communication*

Received date: 10 April 2019
Revised date: 27 July 2019
Accepted date: 3 September 2019

Please cite this article as: Santiago Pascual, Joan Serrà, Antonio Bonafonte, Time-domain Speech Enhancement Using Generative Adversarial Networks, *Speech Communication* (2019), doi: https://doi.org/10.1016/j.specom.2019.09.001

# Time-domain Speech Enhancement Using Generative Adversarial Networks

Santiago Pascual[a,*], Joan Serrà[b], Antonio Bonafonte[a]

[a] *Universitat Politècnica de Catalunya, Barcelona*
[b] *Telefónica Research, Barcelona*

**Abstract**

Speech enhancement improves recorded voice utterances to eliminate noise that might be impeding their intelligibility or compromising their quality. Typical speech enhancement systems are based on regression approaches that subtract noise or predict clean signals. Most of them do not operate directly on waveforms. In this work, we propose a generative approach to regenerate corrupted signals into a clean version by using generative adversarial networks on the raw signal. We also explore several variations of the proposed system, obtaining insights into proper architectural choices for an adversarially trained, convolutional autoencoder applied to speech. We conduct both objective and subjective evaluations to assess the performance of the proposed method. The former helps us choose among variations and better tune hyperparameters, while the latter is used in a listening experiment with 42 subjects, confirming the effectiveness of the approach in the real world. We also demonstrate the applicability of the approach for more generalized speech enhancement, where we have to regenerate voices from whispered signals.

*Keywords:* speech enhancement, audio transformation, generative adversarial network, neural networks.

*Corresponding author.
*Email address:* santi.pascual@upc.edu (Santiago Pascual)

## 1. Introduction

Speech enhancement aims to improve the intelligibility and quality of speech contaminated by additive noise (Loizou, 2013). Its main applications are related to improving the quality of communications in noisy environments. However, we also find applications related to hearing aids and cochlear implants, where enhancing the signal before amplification can significantly reduce discomfort and increase intelligibility (Yang and Fu, 2005). Speech enhancement has also been successfully applied as a preprocessing stage in speech recognition and speaker identification systems (Ortega-Garcia and Gonzalez-Rodriguez, 1996; Yu et al., 2008; Maas et al., 2012). For instance, the later-presented frequency-domain version of our proposed system has already been used as a front end in a speech recognition pipeline (Donahue et al., 2018a).

Most of the current speech enhancement systems are based on the short-time Fourier analysis/synthesis framework, where only the spectral magnitude is treated to remove contaminating artifacts (Loizou, 2013). Recovering the signal is, in that case, a matter of recombining the cleaned-up magnitude with the input phase. This approach is common practice, as it is often claimed that short-time phase is not important for speech enhancement (Wang and Lim, 1982). Nonetheless, other studies show that significant improvements of speech quality are possible, particularly when a clean phase spectrum is known (Paliwal et al., 2011).

Generative adversarial networks (GANs; Goodfellow et al., 2014) are state-of-the-art generative models within the deep learning framework. In this work, we present a GAN for speech enhancement that works with the raw audio signal (Pascual et al., 2017) and aims at more generalized speech enhancement tasks. We first describe our speech enhancement GAN (SEGAN) applied to denoising, together with an extensive exploration of variations that leads to an increase in performance and efficiency. With this step, we find that two key variations are the introduction of learnable skip connections and the reduction of the architecture size by means of larger convolutional strides, which in turn increases the adversarial training stability. We then compare our approach to classic speech enhancement algorithms, such as Wiener filtering and statistical-based methods (Loizou, 2013), and to other deep neural networks working on the frequency domain. A novel application of SEGAN is also presented that goes beyond simple denoising and timewise sample correspondence as a result of several changes in the GAN loss functions. We

first substitute temporal regularization with spectral regularization. Second, we enforce content preservation with the addition of an extra adversarial signal. With these augmentations, we address the regeneration of whispered speech into a more natural and voiced signal. We call this modification a whispered-to-voiced conversion, applicable to the assistance of people lacking vocal folds, potentially after total laryngectomy surgery (Gonzalez et al., 2017a,b).

The article is structured as follows. Sec. 2 is a review of different types of speech enhancement algorithms in our current context. This review is followed by a more detailed review of GANs in Sec. 3. Our model is then introduced in Sec. 4, covering all architectural, configuration, and training details. Sec. 5 explains the experimental configuration. In Sec. 6, we describe several model variations and their objective results, together with a subjective comparison of the best-performing variant against other competitive systems. Sec. 7 presents the application of whispered-to-voiced conversion. Finally, conclusions are discussed in Sec. 8. Code for our approaches[1], deep learning baselines[2], and audio samples for denoising[3] and dewhisper[4] applications are available online.

## 2. Related Work

Classic speech enhancement includes spectral subtraction (Berouti et al., 1979), Wiener filtering (Lim and Oppenheim, 1978), statistics-based methods (Ephraim, 1992) such as the minimum mean squared error (MMSE), and subspace algorithms (Dendrinos et al., 1991; Ephraim and Van Trees, 1995). Neural networks are a recent and successful trend for this task, although they were initially applied in the 1980s by Tamura and Waibel (1988), and later by Parveen and Green (2004). Recent widely used architectures typically work in the spectral domain, as with classic techniques, to learn a regression to the clean spectrum, typically in the form of a denoising autoencoder (DAE; Lu et al., 2013; Xu et al., 2015). Other approaches work by predicting masks with deep neural networks that palliate noisy spectral regions (Narayanan and Wang, 2013; Williamson and Wang, 2017; Wang et al., 2014). Recurrent

[1] `https://github.com/santi-pdp/segan_pytorch`
[2] `https://github.com/santi-pdp/spentk`
[3] `http://veu.talp.cat/seganp`
[4] `http://veu.talp.cat/whispersegan`

neural networks (RNNs) are also used, owing to their success in modeling sequential processes. Research shows that RNNs can predict a better contextualized set of frames or masks (Maas et al., 2012; Weninger et al., 2015, 2014; Erdogan et al., 2015). The use of dropout, postfiltering, and perceptually motivated metrics is also effective. Xia and Bao (2013) propose to use a weighted DAE, altering the mean squared error loss function by assigning weighting factors to each spectral component. Furthermore, Shivakumar and Georgiou (2016) use a loss function that considers the perceptual quality of speech, and Fu et al. (2018) use an intelligibility loss to obtain better scores than those of plain regression losses. Williamson and Wang (2017) use a deep neural network (DNN) in the spectral domain, including the phase, by working with complex masks.

Convolutional neural networks (CNNs) are also known to perform well for locally correlated data, such as speech waveforms or spectrograms. As such, we have used them for one of the first speech enhancement systems working with the raw audio signal (Pascual et al., 2017). Other contemporary studies use deep convolutional structures for this task in the form of regression architectures, such as the work by Park and Lee (2017), who emphasize the need for reduction in model size (typically achievable through CNNs), or the denoising WaveNet (Rethage et al., 2018). Other approaches use improvements in the adversarial setup in the form of a Wasserstein GAN with gradient penalty (Gulrajani et al., 2017; Qin and Jiang, 2018). Moreover, adversarial losses have been used in the speech enhancement field to work without parallel corpora of aligned pairs (Higuchi et al., 2017). The adversarial framework also appeared as a methodology to combine speech enhancement together with automatic speech recognition systems, either in the waveform or the spectral domain (Donahue et al., 2018a; Meng et al., 2018).



## 3. Generative Adversarial Networks

GANs (Goodfellow et al., 2014) are generative models that learn to map samples $\mathbf{z}$ from some prior distribution $\mathcal{Z}$ to samples $\mathbf{x}$ from another distribution $\mathcal{X}$, which is the one of the training instances (e.g., images or audio). The component within the GAN structure that performs the mapping is called the generator network (G), and its main task is to learn a function whose outcomes can imitate some real data distribution. In this way, we can generate novel samples related to those of the training set. Importantly, G

does so not by memorizing input-output pairs but by mapping the data distribution characteristics to the manifold defined in our prior $\mathcal{Z}$. Thus, there is an inherent stochastic component (in this case the sampling from $\mathcal{Z}$) that implies a different outcome for every generated prediction.

Adversarial training is the key component with which G learns to perform the aforementioned mapping. In this configuration, we have another component, called the discriminator network (D), which is typically a binary classifier. Its inputs are either real samples, coming from the dataset, or synthetic samples, entirely made up by G (which in turn imitates real samples). The adversarial characteristic comes from the fact that D has to classify the samples coming from $\mathcal{X}$ as real, whereas the samples coming from G, $\hat{\mathcal{X}}$, have to be classified as synthetic. This condition leads to G trying to fool D, and the way to do so is that G adapts its parameters such that D classifies the G output as real. During back-propagation, D improves at finding realistic features in its input; in turn, G corrects its parameters to move towards the real data manifold described by the training data (Fig. 1). This adversarial learning process is formulated as a minimax game between G and D, with the objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \\ + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right)\right].$$

The previously described model would learn to generate novel samples that could resemble random real points, but it is often interesting to add a conditioning factor that can be used for a specific task. We can thus work with a conditioned version of GANs, where we have some additional information in $G$ and $D$ to perform mapping and classification (see Isola et al., 2017, and rerefernces therein). In our case, we can condition the generation to a contaminated input utterance such that $G$ has to output a clean version of it. The minimax game formulation is then modified to include a conditioning input vector $\tilde{\mathbf{x}}$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x},\tilde{\mathbf{x}}\sim p_{\text{data}}(\mathbf{x},\tilde{\mathbf{x}})}\left[\log D(\mathbf{x},\tilde{\mathbf{x}})\right] + \\ + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}\left[\log\left(1 - D\left(G\left(\mathbf{z},\tilde{\mathbf{x}}\right),\tilde{\mathbf{x}}\right)\right)\right]. \tag{1}$$

Note that $D$ is also receiving the conditioning vector $\tilde{\mathbf{x}}$ such that the information flowing back from $D$ to $G$ during training incorporates tied descriptions of both reality $\mathbf{x}$ (clean signal) and its conditioning reference $\tilde{\mathbf{x}}$ (noisy or corrupted signal).

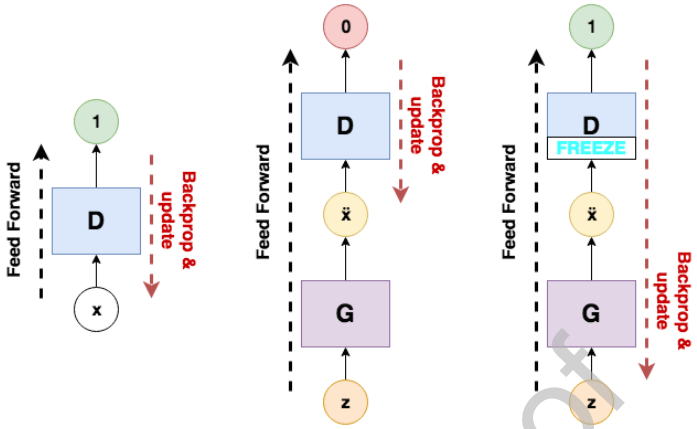


Figure 1: Schema of the GAN training process. First, $D$ back-props a batch of real examples (left). Then, $D$ back-props a batch of synthetic examples that come from $G$ and classifies them as synthetic (middle). Finally, the $D$ parameters are frozen, and $G$ back-props to make $D$ misclassify the examples (right).

There have been a number of improvements to the classifier structure of the discriminator to stabilize the overall adversarial training. These developments make $D$ learn better features, with a better gradient flow in some cases relative to the classical formulation, which in turn improves the training of $G$, as it receives better error signals. The binary classification output in $D$ can suffer from vanishing gradients due to the sigmoid cross-entropy loss used for training. To solve this problem, the least squares GAN (LSGAN) approach (Mao et al., 2017) replaces the cross-entropy loss with the least squares function and an output linear unit, keeping the same binary coding (1 for real and 0 for synthetic). With this replacement, the formulation in Eq. 1 changes to

$$\begin{aligned}
\min_G V(G) &= \frac{1}{2}\,\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}[(D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})-1)^2],\\
\max_D V(D) &= \frac{1}{2}\,\mathbb{E}_{\mathbf{x},\tilde{\mathbf{x}}\sim p_{\text{data}}(\mathbf{x},\tilde{\mathbf{x}})}[(D(\mathbf{x},\tilde{\mathbf{x}})-1)^2]\\
&\quad+\frac{1}{2}\,\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}[D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})^2].
\end{aligned}$$

## 4. Speech Enhancement GAN

### *4.1. Model*

In the enhancement problem, we have an input noisy signal $\tilde{\mathbf{x}}$, and we wish to clean it, thus obtaining the enhanced signal $\hat{\mathbf{x}}$. In our configuration we have, for every noisy signal, its clean reference $\mathbf{x}$ during training. The proposed model follows the conditioned generative adversarial approach described in Sec. 3. We call our model speech enhancement GAN, or SEGAN for short. $G$ is structured as a deep convolutional autoencoder (Fig. 2) that first compresses the input waveform in time with the encoder and then reconstructs a plausible clean version of it with the decoder. Compression is done to discourage learning the identity function in the reconstruction of $\tilde{\mathbf{x}}$. Additionally, this autoencoder design also accelerates the convolution operations in the decimated parts of the structure (shorter sequence lengths involve faster processing) and reduces the memory footprint by using smaller feature maps.

The generator input is the noisy speech signal $\tilde{\mathbf{x}}$, which is projected into an intermediate representation (see below). Its output is the enhanced version $\hat{\mathbf{x}} = G(\mathbf{z}, \tilde{\mathbf{x}})$. As the design of $G$ is exclusively convolutional, there are no fully connected layers nor autoregressive connections. This condition encourages the network to focus on temporally close correlations in the input signal and throughout the whole forward process across layers. Additionally, we note that it is a fast way to perform forward operations, as we process the full signal with one forward operation through the whole $G$. This approach contrasts with that of autoregressive or RNN models, which cannot be parallelized when computing each time step.

In the generation stage, the input signal $\tilde{\mathbf{x}}$ is decimated and expanded featurewise through a number of strided convolutional layers, followed by multiparametric rectified linear units (PReLUs, i.e., learnable activation negative slope per feature channel; He et al., 2015). We choose strided convolutions as they are more stable than other pooling approaches for well-known GAN configurations (Radford et al., 2015). Decimation is implemented until we obtain a condensed representation of a few time samples (in the form of vectors of features), commonly called the thought vector $\mathbf{c}$. This result is concatenated with the generative noise component $\mathbf{z}$, which adds stochastic behavior to the generator predictions $\hat{\mathbf{x}}$ (we use isotropic Gaussian noise for $\mathbf{z}$). The encoding process is reversed in the decoding stage by means of transposed convolutions (sometimes called deconvolutions), followed again by

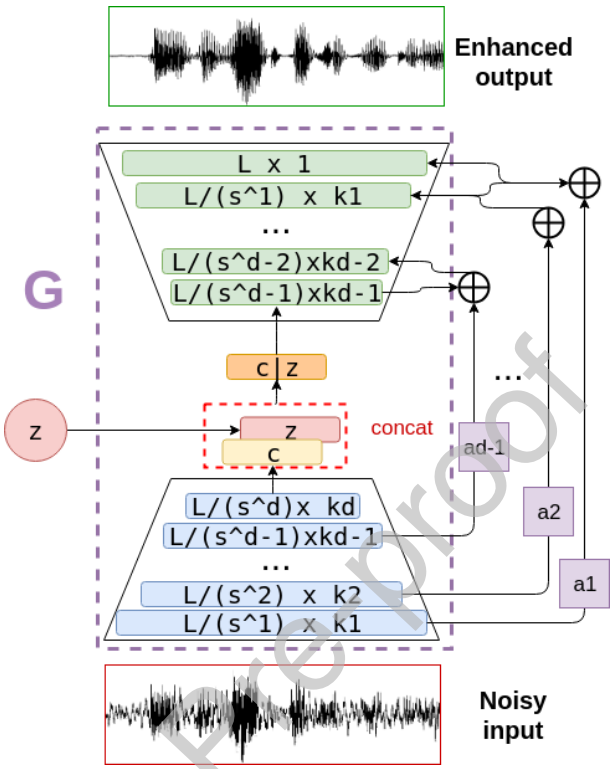


Figure 2: Autoencoder architecture for speech enhancement ($G$ network). Feature maps are depicted in blue and green. The decimation/interpolation factor $s^d$ depends on the stride $s$ and layer depth index $d$. The input waveform length is designated $L$, and the number of kernels/channels at each layer is $k_d$. The right-side arrows denote skip connections, which have a multiplicative scalar factor $a_d$.

PReLUs. The only exception is the last layer, which has a tanh activation to ensure that the output range is between $-1$ and 1. This step is introduced for stability purposes as it avoids exploding gradients, owing to its activation saturating regions, as in deep convolutional GAN (Radford et al., 2015).

The generator $G$ also features skip connections, linking each encoding convolutional layer output to its homologous decoding layer and bypassing the compression performed in the middle of the model (Fig. 2). We do this because the input and output signals of the model share the same underlying structure of natural speech. We hypothesize that low-level details to reconstruct the speech waveform properly could be lost if we force all information to flow through the compression bottleneck. Skip connections can help in

this scenario, directly passing the fine-grained information of the waveform to the decoding stage. Moreover, we observed that skip connections offer a better training behavior, as the gradients can flow deeper through the whole structure (He et al., 2016). Our skip connections contain a multiplicative scalar factor $a_{l,k}$ per signal channel $k$ and layer $l$. Therefore, if we have $k = 16$ channels, after the first encoder layer ($l = 1$) we will have a vector $\mathbf{a}_1 \in \mathbb{R}^{16}$ of amplifying or attenuating factors. These $\mathbf{a}_l$ vectors are learned together with the whole convolutional structure. In this way, the scaling of every feature map is optimized for the end task. At the $j$-th decoder layer input, we merge the (scaled) $l$-th encoder layer with the $j-1$-th decoder layer responses, following either a summation,

$$\mathbf{h}_j = \mathbf{h}_{j-1} + \mathbf{a}_l \odot \mathbf{h}_l,$$

or a concatenation,

$$\mathbf{h}_j = [\mathbf{h}_{j-1}; \mathbf{a}_l \odot \mathbf{h}_l],$$

where $\mathbf{h}_j$ is the output of the $j$-th layer and $\odot$ is an elementwise product along channels. Concatenation gives us slightly better results, but summation can also be competitive and compelling to make the system work with computationally restricted resources, as it requires fewer feature maps than the other option (see Sec. 6).

To complete the GAN structure, we have the discriminator network, which follows the same one-dimensional convolutional structure as the $G$ encoder, hence matching the conventional topology of a convolutional classification network. However, there are a few differences from the $G$ encoder: (1) the discriminator network provides two input channels, (2) it can use some form of batch normalization (Ioffe and Szegedy, 2015) before LeakyReLU nonlinearities of $\alpha = 0.3$, and (3) in the last activation layer, there is a one-dimensional convolution layer with a single filter of width 1 and stride 1. The latter (3) reduces the amount of parameters required for the final classification neuron, which is fully connected to all hidden activations with a linear behavior (no activation function in between). This aspect reduces the amount of required parameters in the last activation from $T \times 1024$ to $T$ with a learnable weighting.

### *4.2. Training*

With the generator $G$ and the discriminator $D$, we then build the adversarial setup, which means that $D$ leaks information to $G$ during backpropagation of what is real and what is synthetic. This way, $G$ can slightly

correct its output waveform towards the realistic distribution, discarding the noisy signals as those are signaled to be synthetic. In this sense, $D$ can be understood as learning some sort of loss for the $G$ output to look real, so that the enhancement must remain faithful to the speech signal and eliminate all the surrounding noise as much as possible. However, in preliminary experiments, we found it convenient to add a secondary regression component to the loss of $G$ to minimize the distance between its generations and the clean examples. This way, the adversarial component can add more fine-grained and realistic results to the regression component. Both losses together were more stable than the separated case.

We chose the $L_1$ norm to be our regularizer, as it has been proven to be effective in the image manipulation domain (Isola et al., 2017; Pathak et al., 2016). Therefore, the $G$ loss becomes

$$\begin{aligned}\min_G V(G) = \frac{1}{2}\,&\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{x}}\sim p_{\text{data}}(\tilde{\mathbf{x}})}[(D(G(\mathbf{z},\tilde{\mathbf{x}}),\tilde{\mathbf{x}})-1)^2] \\ &+\lambda\,\|G(\mathbf{z},\tilde{\mathbf{x}})-\mathbf{x}\|_1,\end{aligned} \tag{2}$$

where $\lambda$ is a hyperparameter that controls the magnitude of the regression component. We set $\lambda$ to 100 after observing a better minimization trend correlated with signal quality. If $\lambda$ is set to a smaller value, the $L_1$ term oscillates erratically. If it is set to a larger value, $G$ behaves as a simple regressor. At approximately 100, this regularization term helps stabilize the training and yields favorable results (which is expected on a purely signal denoising task). However, having an $L_1$ regularization term can be a limitation when we have misalignment in input/output pairs, as it forces every sample of the output to match with the corresponding sample of the input. We did not encounter this problem when removing additive noise, but we had to replace this term when dealing with speech reconstruction (see Sec. 7).

In terms of input data for $D$, and contrasting to typical adversarial training configurations, our configuration does not check whether a chunk is real or synthetic. Instead, training works with pairs of chunks that make real or synthetic targets as follows: a real pair is composed by a clean and a noisy signal $(\mathbf{x},\tilde{\mathbf{x}})$ and a synthetic pair is composed by an enhanced and a noisy signal $(\hat{\mathbf{x}},\tilde{\mathbf{x}})$. This is why $D$ needs a *stereo* input: it classifies the comparison between both chunks as being real or synthetic. The training data were obtained by sliding a window of 16,384 samples (approximately 1 s at 16 kHz) every 500 ms from every training waveform. To study the variations on our architecture, models train for 100 epochs with a batch size of 300. A

validation set of another 300 segments with maximum variability (different speakers than those in training, different noises, and different SNRs) is used to find reasonable plateaus in COVL and SSNR metrics (Sec. 5.3).

### *4.3. Generation*

As $G$ is a fully convolutional structure, we can forward any chunk size $L$ at test/generation time, with the only restriction being that the size be a multiple of the decimation factor $\Delta$ of the encoder. This means that we have to pad sequences with $P$ zeros in some cases to fulfill $N = \frac{L+P}{\Delta} \in \mathbb{Z}$, so that we recover $L+P$ samples in the decoder output, to finally remove the leftover $P$ values and retain our original $L$ samples in our region of interest. When it comes to training, however, $D$ has a fully connected output classification layer, which requires us to use fixed-size chunks.

At test/generation time, the difference between concatenating individually processed chunks of 1 second and processing any length T through $G$ was objectively negligible. Hence, for long signals where intermediate network activations do not fit in memory (neither GPU's nor RAM), we can chunk without overlap, and by sliding $G$ with the same $\mathbf{z}$ through the chunks, we can reconstruct with a concatenation.

## 5. Experimental Configuration

### *5.1. Data*

To evaluate the effectiveness of our approach, we employ the clean speech in the VCTK Corpus (Veaux et al., 2016) and the noises from the Demand dataset (Thiemann et al., 2013), together with some extra synthesized noises following the structure and scripts of Valentini-Botinhao et al. (2016). We choose to generate these data ourselves, based on publicly available datasets with a massive amount of speakers because, this way, we can increase the pool of available speaker variability following the same SNR and noise variation structure as in Valentini-Botinhao et al. (2016). We have 109 available speakers in VCTK, out of which we split into 80 for training, 14 for validation, and 15 for testing. We force different speakers per split to study the generalization of the enhancement algorithm to unseen speakers.

To further lessen the intersection between splits, we run preprocessing to look for the least possible intersection in terms of textual contents between them. We find a total of 44,085 text files in the corpus, but with simple preprocessing, we obtain approximately 14,000 unique ones at the sentence

level. We process the text files by lowercasing and eliminating any carriage return characters, punctuation signs, and repeated spaces. This way, we obtain strings that can be compared literally among speakers (even though some strings can still have same spoken contents but slightly different text files, with some spontaneous missing determinants such as "the"). Based on this simple rule, we select the 15 speakers that have minimum text intersection with others for testing. The validation set is made of 14 speakers within the remaining pool of 94 available ones after test selection. We also retain gender balance to remain consistent in each subcorpus.

The noise conditions imposed with the abovementioned structure and scripts are 10 different noises with 4 SNR levels each for training and 5 different noises with other 4 SNR levels each for testing. The SNR conditions were $\{0, 5, 10, 15\}$ dB for training and $\{2.5, 7.5, 12.5, 17.5\}$ dB for testing. The noises used were (1) synthetic babble: many speakers in background; (2) real cafeteria: a busy office cafeteria; (3) real car: in a private passenger vehicle; (4) real kitchen: inside a kitchen preparing food; (5) real meeting; (6) real metro: a subway; (7) real restaurant; (8) synthetic ssn: white noise low-pass filtered; (9) real station; and (10) real traffic: a busy traffic intersection; for training/validation, and (1) real bus; (2) real cafe: the terrace of a cafe at a public square; (3) real living: inside a living room; (4) real office; and (5) real square: a public town square with tourists; for testing (noise types were randomly selected).

### *5.2. Baselines*

We compare our model with two sets of baselines: (1) classic methods that do not require training parameters and (2) two deep learning methods that work in the spectral domain. Regarding the classic methods, we used a Wiener filter together with a statistical model based on the LogMMSE estimator. Both are taken from Loizou (2013).

The deep learning methods are based on discriminative learnable non-linear mappings. First we have models mapping noisy spectrum frames to clean spectrum frames. Based on Xu et al. (2015) and the modifications of the deep neural network baseline by Fu et al. (2017), we first build a deep neural network (DNN) with fully connected units, where we inject $C$ input log-power spectral frames and obtain a single clean one. Consequently, we have a context window for which we clean up

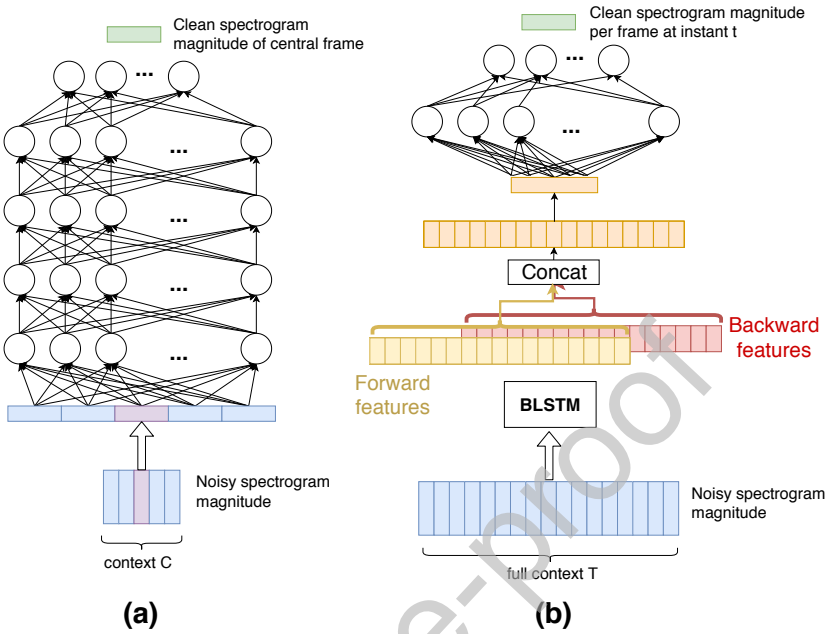


Figure 3: (a) LPS-DNN: Deep neural network baseline mapping of a context window of $C$ log-power spectral frames to the central clean frame. (b) LPS-BLSTM: Bidirectional long-short term memory recurrent neural network that maps the full input noisy sequence to the clean one. The output of the BLSTM (forward and backward extracted features) is fed to an additional multi-layer perceptron to fullfil a final transformation and dimension adaptation sharing weights through time.

the central frame (Fig. 3). The structure of the network is a stack of 4 hidden layers of 1024 units each and an output layer that projects to the proper dimensionality to match the number of frequency bins $F$ of our signal. We refer to this model as log-power spectrum DNN (LPS-DNN). Every hidden layer is a stack of an affine transformation, followed by a multiparametric PReLU activation and batch normalization.

The FFT resolution remains for all experiments at 512, so that we address $F = 257$ bins. We then consider $L_1$ and $L_2$ losses and variations of $C$ to check the regime of values in which we have a competitive baseline.

We also implement a bidirectional LSTM network (BLSTM; Hochreiter and Schmidhuber, 1997) for its known good modelling capacity for se-

quential problems like this (Maas et al., 2012; Weninger et al., 2015, 2014; Erdogan et al., 2015). Our BLSTM has 650 cells, followed by a multi-layer perceptron to perform a final transformation and dimension adaptation. This last module's parameters are shared at all time-steps, and the hidden layer is of size 1024, following the LPS-DNN output structure. This network is designed to be comparable in terms of parameters to those of the LPS-DNN with $C = 1$, where only the sequential processing structure is changed. We also perform the $L_1$ and $L_2$ loss variations. We find that the LPS-BLSTM models require more careful tuning than the LPS-DNN given their tendency to stop learning because of gradient propagation issues if activations and gradients do not have the proper magnitudes (Hochreiter and Schmidhuber, 1997). For this reason, we normalize the first two statistical moments of the inputs and apply gradient clipping to stabilize and promote the proper learning during back-propagation through time (Pascanu et al., 2013), which we observe to be beneficial in this case.

Finally, we also make a fully convolutional auto-encoder structure trained as a plain $L_1$ regression, hence decoupling the adversarial component from the system presented in Sec. 4. The specific configuration of this model is the best one resulting from the ablation study performed in Sec. 6.1, so that only the adversarial component is removed. This is an interesting way to assess the effect of the adversarial component for the considered tasks.

All models were trained with the Adam optimizer (Kingma and Ba, 2015) with default parameters as in PyTorch version 0.4.1 (Paszke et al., 2017). They are trained with all noise types, SNR conditions, and speakers (Sec. 5.1). Note that these approaches do not use any supervision such as speaker identity or noise type. This way, we expect them to generalize to the different kinds of noises and speakers, which we also do with our model. These baselines are competitive counterparts to our waveform-based model. Specially the LPS ones as they work with more condensed information, with a prelocation step we perform that focuses on the spectral magnitude, where additive noise can be detected and removed easily.

### *5.3. Objective Metrics*

We evaluate the quality of the enhanced speech with a set of well-known objective metrics, which serve as tools to obtain an estimation on how well

the models work. All of them compare the enhanced signal with the clean reference of 4,432 test set files. They have been computed with our Python reimplementation of the algorithms in Loizou (2013), which were available at the publisher website. The metrics, their meaning, and their range of values are as follows:

- PESQ: Perceptual evaluation of speech quality using the wide-band version recommended in ITU-T P.862.2 (ITU, 2007).
- CSIG: Mean opinion score (MOS) prediction of the signal distortion attending only to the speech signal (Hu and Loizou, 2008).
- CBAK: MOS prediction of the intrusiveness of background noise (Hu and Loizou, 2008).
- COVL: MOS prediction of the overall effect (Hu and Loizou, 2008).
- SSNR: Segmental SNR (Scalart and Filho, 1996).
- STOI: Short-time objective intelligibility (Taal et al., 2010, 2011).

The PESQ metric ranges between -0.5 and 4.5. MOS regression metrics (CSIG, CBAK, and COVL) take values between 1 and 5. SSNR, in dB, is in the range $[-10, 35]$, as we trim it following the abovementioned implementation. STOI takes values in the range of 0 to 1. For all metrics, the higher the score is, the better the speech quality and the intelligibility.

## 6. Results

In the following, we make two blocks of analyses. In the first block, we conduct an ablation study of different SEGAN configurations and structures, departing from our first version introduced in Pascual et al. (2017). New configurations allow us to obtain an improved version of SEGAN, SEGAN+, which we take as our current best model. In the second block of analyses, we make performance comparisons between SEGAN/SEGAN+ and baseline systems, which comprises both objective and subjective evaluations.

### *6.1. Model Variations*

The variations introduced in the first block of experiments to improve SEGAN are the encoder/decoder stride size, encoder/decoder kernel size, optimizer, normalization schemes, enhancement with $\mathbf{z}$, and skip connections design. All variation results are shown in Table 1, where model variants have an identifier that follows a tree development, from V1 (first SEGAN) toward the latest leaves in the V1.2.8.x level. We highlight the best node (SEGAN+) of the tree in bold. We follow these different node details in the following sections, where a set of takeaways emerge:

- It is important to have an aggressive decimation factor per encoder stage while maintaining a large kernel width. This approach allows for an efficient architecture that, as a result of the large receptive field, maintains satisfactory performance.
- Some normalization mechanism, either spectral normalization ($G$ and $D$) or batch normalization ($D$), is essential for the correct training of the system. Both increase performance in a similar way, allowing for better training gradient flows for a deep structure such as the one we consider.
- Learnable skip connections are a significant improvement on the $G$ architecture. Using a scalar factor per hidden feature allows for importance filtering of feature maps from encoder to decoder. This approach provides better results with the same stability and similar gradient propagation as regular skip connections.
- The latent vector $\mathbf{z}$ is not clearly used as a generative component in noise removal, but we find that it helps as a regularization factor of $G$ (without any additional requirements of dropout, batch normalization, or weight magnitude restrictions).

In the following subsections, we comment each variation in detail.

Table 1: Objective performance for different architecture variations. SEGAN is the first approach we developed in Pascual et al. (2017) but is evaluated over the new dataset (V1). SEGAN+ is the new best-performing model out of the different variations (V1.2.8). For both COVL and SSNR metrics, higher is better. Letters $\eta$ and $\omega$ denote the learning rate and kernel width, respectively.

| Model | Description | COVL | SSNR |
|---|---|---|---|
| V1 (SEGAN) | SEGAN first version with a stride of 2, a kernel width of 31, and batch norm in $D$. | 2.77 | 5.15 |
| V1.1 | V1 made smaller and narrower, with a stride of 4 and a kernel width of 11. | 2.58 | 4.38 |
| V1.2 | V1 made smaller with the same kernel sizes, with a stride of 4 and a kernel width of 31. | 2.89 | 6.65 |
| V1.2.1 | V1.2 with spectral normalization in $G$ and $D$ and $\eta_G = 10^{-4}$, $\eta_D = 4 \cdot 10^{-4}$. | 2.33 | 6.42 |
| V1.2.2 | V1.2 with spectral normalization in $G$ and $D$ and $\eta_G = 10^{-4}$, $\eta_D = 4 \cdot 10^{-4}$, and no batch norm in $D$. | 2.72 | 6.56 |
| V1.2.3 | V1.2 with Adam and no batch norm in $D$. | 2.10 | 3.35 |
| V1.2.4 | V1.2 with spectral normalization in $G$ and $D$ and $\eta_G = 10^{-4}$, $\eta_D = 4 \cdot 10^{-4}$, Adam, and no batch norm in $D$. | 2.88 | 6.59 |
| V1.2.5 | V1.2 with no batch norm in $D$. | 2.00 | 3.75 |
| V1.2.6 | V1.2 with Adam. | 2.73 | 6.84 |
| V1.2.7 | V1.2 with convolutional skip connections. | 2.73 | 3.30 |
| **V1.2.8 (SEGAN+)** | **V1.2 with learnable scalar skip connections initialized at 1.** | **3.00** | **7.05** |
| V1.2.8.1 | V1.2.8 with skip connections initialized at 0. | 2.89 | 6.83 |
| V1.2.8.2 | V1.2.8 with summation merge of feature maps. | 2.83 | 6.82 |
| V1.2.8.3 | V1.2.8 skipping post-activation feature maps. | 2.90 | 6.04 |
| V1.2.8.4 | V1.2.8 without $\mathbf{z}$ vector. | 2.20 | 4.19 |
| V1.2.8.5 | V1.2.8 without biases. | 2.88 | 7.11 |
| V1.2.8.6 | V1.2.8 modifying kernel widths: $\omega_{G\text{enc}} = 31$, $\omega_{G\text{dec}} = 4$, and $\omega_D = 31$. | 2.61 | 6.37 |
| V1.2.8.7 | V1.2.8 modifying kernel widths: $\omega_{G\text{enc}} = 31$, $\omega_{G\text{dec}} = 4$, $\omega_D = 31$, and no biases. | 2.83 | 5.96 |

*6.1.1. Encoder/decoder stride and kernel sizes*

In this first level of experiments, we determine the effectiveness of increasing kernel stride in terms of both stability and performance, in addition to the degradations in performance associated with smaller kernel widths (V1.1 and V1.2, Table 1). This condition also makes $G$ more efficient, yielding a generation process that is 1.7 times faster than real time on a CPU and 17 times faster on a GPU.

Sec. 4 introduces SEGAN as a flexible deep convolutional design. The first SEGAN proposal (V1) is composed of convolutions/deconvolutions of stride 2 and kernel width 31. The feature map configuration of the $G$ network is as follows: $16384 \times 1$, $8192 \times 16$, $4096 \times 32$, $2048 \times 32$, $1024 \times 64$, $512 \times 64$, $256 \times 128$, $128 \times 128$, $64 \times 256$, $32 \times 256$, $16 \times 512$, and $8 \times 1024$. This configuration is mirrored in the decoder to go back to $16384 \times 1$ resolution, with some possible doubled feature channels if we use concatenative skip connections (Sec. 4).

One of the goals of our design is its speed, and decimation is a key factor to increase speed in a fully convolutional setup. An initial step we take is to reduce the size of model, hence increasing its computational efficiency, by means of increasing the stride factor from 2 to 4. After changing the stride to 4, we reduce the amount of layers from 22 to 10 such that we obtain the feature map structure $16384 \times 1$, $4096 \times 64$, $1024 \times 128$, $256 \times 256$, $64 \times 512$, and $16 \times 1024$ in the $G$ encoder. Our first interest was reducing the model depth itself while maintaining the quality, but we found a quality increase, and we hypothesize that this factor might be related not only to less depth but also to the change in decimation, as we observe a change in high-frequency artifacts appearing in the $G$ output with this structural change. Recently, the aliasing effect increasing with convolutional pooling was shown to degrade classification tasks with a waveform injected into the network (Gong and Poellabauer, 2018). Nonetheless, it remains unclear how this aliasing affects the quality of waveform generative models (Donahue et al., 2018b), as it may be used to reconstruct missing frequency bands when upsampling from the latent space in GAN frameworks.

*6.1.2. Normalization schemes*

After experimenting with different normalization schemes, we determine how important they are to stabilize the adversarial training. Hence, either batch normalization in $D$ or spectral normalization in both networks ($G$ and $D$) is required to obtain training stability. Nonetheless, they should not be

applied jointly because the performance would degrade. These observations are shown in results V1.2.1–6 (Table 1), where we vary optimizers and normalization schemes.

Whenever we do not use any form of normalization (V1.2.3 and V1.2.5), we obtain more unstable results that lead to lower objective scores, particularly in terms of SSNR, as outputs are quite noisy. Hence, unless we use some form of normalization somewhere in the full GAN structure, either training diverges or the results are noisy and of poor quality. Moreover, we encountered no substantial difference between using virtual batch normalization as we did originally (Pascual et al., 2017) or plain batch normalization while reproducing SEGAN on the current data (V1). Thus, we use plain batch normalization for the sake of simplicity in all our current experiments. Spectral normalization, a promising technique for conditioned generator structures, is based on upper bounding gradient magnitudes (Zhang et al., 2018). Nonetheless, we could not obtain a better result than the one we had with plain $G$ and batch normalization in $D$.

*6.1.3. Optimizer*

We also find that both Adam and RMSprop (Tieleman and Hinton, 2012) optimizers are effective and yield a stable training across the different configurations with varying learning rates (V1.2.1–6, Table 1). We depart from V1, with small and balanced learning rates $\eta_G = \eta_D = 5 \cdot 10^{-5}$, and we also implement the recent two-timescale update rule (TTUR; Heusel et al., 2017). TTUR is a promising schedule to emulate a discriminator that is updated more often than the generator by simply applying a scaled ratio $\frac{\eta_D}{\eta_G} = 4$, thus ensuring $\eta_D = 4 \cdot 10^{-4}$ and $\eta_G = 10^{-4}$ (Heusel et al., 2017). Even though we achieve competitive results with TTUR (even better than those of V1), we discard them because the results are not better than that of V1.2. We therefore continue using RMSprop with $\eta_G = \eta_D = 5 \cdot 10^{-5}$.

*6.1.4. Skip connections*

We see that including skip connections with a simple learnable scalar boosts the performance of the system. Skip connections facilitate gradient flow, while learnable scalars are trained to determine which level of detail from the encoder layers is shuttled to the decoder layers. We also found that skip connections in $G$ to help stabilize the training process (so much that if we try to train the system without them, it collapses). In V1, we have the simplest skip connections possible: they forward the feature map through an

identity function to shuttle features, and gradients flow back and reach the deepest part of the structure. We decided to make them learnable such that the optimization process can weight their importance independently because they can act as pseudo-attention mechanisms of what levels of features are more important to be shuttled in the decoding process. Experiment V1.2.8 shows the effectiveness of this approach, surpassing the performance of V1.2.4 (Table 1).

Following the criteria that learnable skip connections can enable additional processing that helps the decoding stages, we also tried configuring them as convolutional layers of kernel width 11. This approach is expected to allow them to transform certain filter bands or shift subsignals temporally in the hidden layers, as much as the kernel width. However, the result of this scheme was not positive (V1.2.7). We hypothesize this is due to the possible introduction of noisy transformations when shuttling the data and to the fact that phase transformations are not well indicated for the task of denoising the speech. Overall, we suggest that convolutional skip connections could be useful in future tasks if we have strong misalignments between input and output signals, so that these connections can operate with signal shifts from encoder to decoder.

In addition to determining that learnable scalar skip connections give us the best result so far, we experiment with two different initialization weights on them: 0 and 1. We reach the conclusion that 1 (V1.2.8) is better than 0 (V.1.2.8.1). This result is an intuitive one, given the issues in gradient and data flow provoked having no connections at the beginning. We also try the summation merge described in Sec. 4 (V1.2.8.2), which is ultimately worse than both concatenation alternatives with different initialization schemes. Still, the use of this scheme might be of interest for running the system in environments where memory and/or computing power is restricted, as having fewer feature maps (and thus parameters) can be important. Finally, we also check what happens if we pick the feature activations after the PReLU (V.1.2.8.3) instead of prior to it (V1.2.8) in the encoder to shuttle them up. These are injected into the input of each mirrored decoder layer as before. Performance degrades in this case, thus indicating the superiority of the linear projection before the activation for the skip connection.

#### *6.1.5. Latent* $\boldsymbol{z}$

We find it beneficial to have a latent vector $\mathbf{z}$ at the core of the $G$ structure, which yields better enhancement performance due to a possible regu-

larization effect in the denoising task. In the GAN context, $\mathbf{z}$ serves as a stochastic element to make novel samples at each inference, thus providing generative characteristic to $G$. Our first intention to place it in $G$ is because the enhanced signal is a regeneration of the noisy one. However, the preliminary hearing results of $\mathbf{z}$ suggest that it only minimally affects any hearable structure in the output (with the same input noisy signal sounding similar to different $\mathbf{z}$), which was already noted in the research of conditional generators when incorporating GANs (Isola et al., 2017). Nonetheless, when we remove $\mathbf{z}$, we systematically find a worse performance in all objective metrics (V1.2.8.4, Table 1). We hypothesize that this effect is related to some form of overfitting when we lack this noise. Further training checkpoints yielded similar performances, including the checkpoint with minimum validation error in COVL and SSNR.

Although $\mathbf{z}$ has a reduced relevance as a generative component in the speech denoising application, it can become a key piece in a speech restoration task. In Sec. 7, we apply SEGAN+ on a more difficult signal regeneration task, specifically, to construct pitch contours from damaged, silent speech. In this new task, we empirically observed the generation of different but plausible pitch contours with the same input. An example of this approach is shown in Fig 5.

*6.1.6. Biases*

After experimenting with the introduction and absence of biases in the full convolutional structures of $G$, we choose to maintain them as they give a higher peak performance in perceptual objective results. Nevertheless, it is worth discussing the importance of not having biases and considering this feature for future implementations. The intuition behind the absence of bias is that if we have pure silence in the input, we should have pure silence in the output. This condition arises only if we have a multiplicative interaction within the network with a zeroed-out signal, which is guaranteed if we do not have bias terms in our convolutions. This variation (V1.2.8.5) shows a slightly better SSNR than V1.2.8, but COVL shows a degradation (Table 1).

*6.1.7. Transposed Convolutions*

We also note the importance of having large kernel widths even in the decoder of the generator. We tried to reduce them to avoid overlapping in the transposed convolution operations, owing to high-frequency artifacts that appear in the output waveform. These could be related to the checkerboard ar-

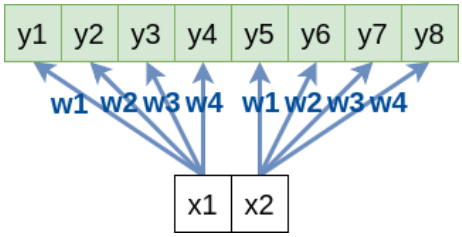


Figure 4: One-dimensional transposed convolution ('deconvolution') with both kernel width and stride equal to 4. The same kernel with weights $\mathbf{w}_i \in \mathbb{R}$ with $i \in [1, 4]$ (in a one-channel input case) does not overlap itself with neighboring samples.

tifacts appearing in image generation transposed convolutions (Odena et al., 2016), something already observed in recent GAN-based speech generation systems (Donahue et al., 2018b). A plausible mechanism to remove them is working with non-overlapping interpolated segments.

We thus try to reduce the kernel width in the decoder layers in an attempt to make non-overlapping interpolations. Fig. 4 shows a schematic of the transposed convolution concept with a non-overlapping form. If we have a kernel width larger than the stride, information between inputs would be mixed in the output of the layer (in the example figure samples, $y_i$ with $i > 4$ would also depend on $x_1$). Intuitively, for a learnable interpolation in the decoder, a non-overlapping upsampling could suffice, provided that the encoder has sufficient capacity to extract a representation with a large receptive field over the input signal. Nevertheless, we find empirical evidence of a worse performance, particularly in terms of SSNR (compare V1.2.8 with V1.2.8.6–7).

### *6.2. Comparative Results*

We now report the results of the aforementioned second block of experiments, comparing SEGAN (V1) and SEGAN+ (V1.2.8) with the considered baselines (Sec. 5.2). We first report objective performance, assessed on a held-out split (Sec. 5.1). Next, we report the results of a subjective preference test, based on the mean opinion scores (MOSs) of 42 subjects.

#### *6.2.1. Objective Performance*

Table 2 shows the comparison between SEGAN, SEGAN+ and the considered baselines. First, we observe that all deep learning baselines are

over the classic baselines, specially for CBAK and COVL. Nonetheless, the SSNR of the LogMMSE is much better than the one of the spectral deep learning baselines: the SSNR of the LPS-DNN and LPS-BLSTM systems is actually at the level of the Wiener filter or below. The LPS-BLSTM approaches achieve better perceptual scores, but do not reach the level of the LPS-DNN systems in this setup. In terms of PESQ and MOS-like metrics, LPS-DNN and LPS-BLSTM systems generally perform better than other systems. For LPS-DNN, increasing the context $C$ actually helps the DNN, as expected. We also observe that there is a slight difference between using $L_2$ and $L_1$ losses in spectral deep learning baselines, but both behave comparably.

It is notorious that the speech enhancement auto-encoder (SEAE+) is objectively comparable to SEGAN+ across perceptual metrics. Nonetheless, CBAK and SSNR show some benefit on using the adversarial component to reduce the intrusiveness of background noise. This result is reasonable for a purely denoising task, where it suffices to remove noisy components as in a regression problem, as stated by Donahue et al. (2018a). However, it does not suffice for other applications that require better generative characteristics as in reconstruction (see Sec. 7).

In terms of STOI, we observe comparable values between all approaches, with SEGAN+ presenting the best average, suggesting a better resulting intelligibility over all presented models. We also observe that SEGAN+ is superior to all other systems in terms of SSNR, which is related to removing more noise, although it has slightly worse PESQ and MOS-like metrics than the DNN-C7 baselines. We suspect that this result is related to the generative capability of the system: the network regenerates a signal that sounds plausible, but such signal still differs from the original one used for evaluation. Another possible source of trouble is high-frequency artifacts that we identify listening to some samples during model development. Such artifacts can introduce accentuated distortions that lower model scores.

*6.2.2. Subjective Performance*

Objective evaluations such as the ones in the previous section are useful indicators, comparing spectral distortions and noise against clean speech levels. However, such evaluations are not completely fair when we face the generation of new data that can include audible artifacts noticeable to humans but not accounted for by the metric. In addition, if the regenerated

Table 2: Objective evaluation results with the considered baselines. The $L_1/L_2$ prefix of DNNs and LSTMs describes the regression loss used, and the $C$ value describes the amount of context frames. For all metrics, higher is better.

| Model | CSIG | CBAK | COVL | PESQ | SSNR | STOI |
|---|---|---|---|---|---|---|
| Noisy | 3.28 | 2.28 | 2.56 | 1.92 | 0.03 | 0.74 |
| Wiener | 2.91 | 2.43 | 2.43 | 2.13 | 3.32 | 0.73 |
| LogMMSE | 3.16 | 2.67 | 2.64 | 2.27 | 5.00 | 0.72 |
| $L_2$-DNN-C1 | 3.82 | 2.70 | 3.02 | 2.26 | 2.98 | 0.70 |
| $L_2$-DNN-C7 | **3.98** | 2.83 | **3.22** | **2.47** | 3.22 | 0.73 |
| $L_1$-DNN-C7 | 3.95 | 2.81 | 3.17 | 2.42 | 3.38 | 0.72 |
| $L_2$-BLSTM | 3.75 | 2.65 | 2.96 | 2.21 | 2.59 | 0.71 |
| $L_1$-BLSTM | 3.82 | 2.69 | 3.01 | 2.24 | 2.84 | 0.70 |
| SEGAN | 3.52 | 2.69 | 2.77 | 2.10 | 5.15 | 0.73 |
| SEAE+ | 3.66 | 2.84 | 3.00 | 2.42 | 5.00 | 0.73 |
| SEGAN+ | 3.66 | **2.97** | 3.00 | 2.37 | **7.05** | **0.75** |

signal differs from the ground truth in terms of amplitude, phase, or other properties that make it intelligible and natural but not an exact fit, objective scores identifying exact matches between low-level properties can lead to misleading results. For these reasons, a subjective test was conducted to assess, with an averaged opinion among many people, how the system performs with regard to the regeneration of plausible speech that resembles a clean reality. For this test, we select the subset of best objectively performant systems per category to compare against SEGAN+. We take the LPS-DNN and LogMMSE as best representatives of the deep learning (spectral) and classic groups, and also maintain the baseline SEGAN and Wiener used in Pascual et al. (2017)as incremental references.

The test was taken by 42 subjects. Each subject was presented with 8 utterances, with each utterance being enhanced by 6 systems. Thus, each user had to rate $8 \times 6 = 48$ audio samples. For each audio, we asked participants to give a MOS rate regarding (a) how intrusive was the background noise (BCK: 5–Not noticeable, 4–Slightly noticeable, 3–Noticeable but not intru-

Table 3: Subjective evaluation results comparing the considered systems. BCK stands for background noise removal and SPE for speech distortion introduced by the system (see Sec. 6.2.2). For both values, higher is better. Each cell shows the mean of each system (standard deviation in parentheses).

| Model | BCK | SPE |
|---|---|---|
| Noisy | 2.84 (1.10) | **4.59 (0.81)** |
| Wiener | 3.19 (1.12) | 4.47 (0.87) |
| LogMMSE | 3.43 (1.08) | 4.44 (0.93) |
| $L_1$-DNN-C7 | 4.26 (1.08) | 4.12 (1.22) |
| SEGAN | 4.24 (1.01) | 4.11 (1.21) |
| SEGAN+ | **4.27 (1.09)** | 4.21 (1.12) |

sive, 2–Somewhat intrusive, and 1–Very intrusive) and (b) how much speech was distorted (SPE: 5–Not distorted, 4–Slightly distorted, 3–Somewhat distorted, 2–Fairly distorted, and 1–Very distorted). Table 3 shows the results for these metrics.

We first focus on the background noise being removed (BCK). We can confirm the incremental gap from Noisy to Wiener and from Wiener to LogMMSE. After LogMMSE, we have the three deep learning systems falling within a comparable range of values, with SEGAN+ achieving a marginally better BCK. In terms of the amount of speech distortion (SPE), we observe a detrimental gap in performance from Noisy to Wiener and LogMMSE, and then the three deep learning systems. Overall, we can understand this result as a trade-off between how much noise they remove and how much speech they destroy. Notably, SEGAN+ remains better than the other two deep learning options for this score, although its performance lies under the classic baselines, as expected, because it clears more intrusive noise as shown in the BCK metric. A conclusion from this result is that with the current subjective results, SEGAN+ seems more selective destroying intrusive signals and that improvements on SEGAN+ make it perform better than the original SEGAN both objectively and subjectively.

## 7. Towards More General Speech Enhancement GANs

In this section, we explore the enhancement capabilities of SEGAN+ beyond the specific task of denoising (i.e., eliminating additive noises of many kinds). An important set of enhancement applications are those that directly affect the speech, allowing for the recovery of more natural spoken utterance out of a damaged one. As a first step in this direction, we explore the application of whispered-to-voiced speech conversions. We refer to this conversion as dewhispering or voicing of the speech signal. Importantly, whispered speech can be uttered on purpose but is also expressed by people suffering from disease or trauma that manifests as aphonia (e.g., patients after a total laryngectomy). The dewhispering process is typically performed in the spectral domain or with vocoder features, similarly to the denoising case. Once these features are obtained, some statistical models such as Gaussian mixture models (Toda et al., 2008; Nakamura et al., 2011, 2012) or DNNs (Gonzalez et al., 2017a) are applied to reconstruct corrupted/missing components, and then, features are reverted to the time domain.

Our whispered utterances are obtained with an articulator motion capture device (Fagan et al., 2008) that monitors the movement of the lips and tongue, tracking the magnetic field generated by small attached magnets. Then, an existing synthesis module generates speech from articulatory data by means of an RNN model trained on parallel articulatory-to-speech samples (Gonzalez et al., 2017a,b). The speech produced by this system has a reasonable quality but sounds monotonous and robotic, owing to limitations when estimating the pitch (i.e., the capturing device does not have access to any information about the glottal excitation). Hence, we can use the RNN to generate a reconstructed whispered speech out of articulatory data while discarding the predicted pitch and then apply SEGAN+ as an enhancement over it. SEGAN+ recovers a more natural sounding speech with a whispered input such that it must implicitly generate pitch curves with proper intonations embedded in the waveform. Here, we compare our model against the existing system that employs the RNN-based architecture to regress pitch and performs a vocoder-based synthesis (Gonzalez et al., 2017b).

### *7.1. Whispered SEGAN*

Importantly, because of the data acquisition and synthesis procedures, small temporal misalignments remain between the input and output records

(i.e., whisper and natural speech differ in length and are not accurately parallel). Therefore, the original SEGAN version is not immediately effective when receiving the new data, particularly because the $L_1$ regularization loss is restricted to work when the output is fully aligned with the input. Similarly, we also expect an $L_1$ auto-encoder to not be effective in the regeneration of missing components, as stated in Sec. 6.2.1. Nonetheless, we consider the SEAE+ architecture in this setup too, as a standalone reconstruction system.

The amount of data we have to carry out this experimentation is 30 min of training utterances plus 3 min of test utterances[5] (it is important to note the large gap in terms of amount of data to train on, from SEGAN/SEGAN+ models, which handle 32 h, relative to this small set of 30 min). We noticed that this data shortage has two main effects: (1) it introduces artifacts at many frequencies, particularly the high ones, and (2) intelligibility is sometimes lost in the reconstruction phase. Hence, we had to make two reformulations to SEGAN+ to adapt it to the current task. We denote this reformulated version as WSEGAN to specify its applicability to dewhispering.

First, time-domain regularization is removed from the loss of $G$, and we use only the power loss as a regularizer (Oord et al., 2017). In this way, we try to mitigate the allocation of energy in non-speech-like frequency bins and thus reduce the aforementioned artifacts. We also add a denoising SEGAN+ processing system on top of WSEGAN to remove erratic artifacts and corrupted speech segments, which acts specifically on silence regions. Second, we introduce a new adversarial loss that enforces content preservation between the input and the output of $G$. More specifically, a synthetic signal (0 in the LSGAN binary coding, Sec. 3) is triggered whenever we have the current clean reference signal $\mathbf{x}$ and another randomly chosen clean signal $\mathbf{x}_r$. Both signals are clean and look natural, and the only difference is the content mismatch; thus, $D$ must learn that mismatched information is not

[5]The corpus contains a single English male speaker that recorded a random subset of the CMU Arctic corpus (Kominek and Black, 2004).

realistic. With these two changes, the WSEGAN loss becomes

$$\begin{aligned}\min_D V(D) &= \frac{1}{3}\,\mathbb{E}_{\mathbf{x},\tilde{\mathbf{w}}\sim p_{\text{data}}(\mathbf{x},\tilde{\mathbf{w}})}[(D(\mathbf{x},\tilde{\mathbf{w}})-1)^2]+\\ &+\frac{1}{3}\,\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{w}}\sim p_{\text{data}}(\tilde{\mathbf{w}})}[D(G(\mathbf{z},\tilde{\mathbf{w}}),\tilde{\mathbf{w}})^2]\\ &+\frac{1}{3}\,\mathbb{E}_{\mathbf{x},\mathbf{x}_r\sim p_{\text{data}}(\mathbf{x})}[D(\mathbf{x},\mathbf{x}_r)^2]\\ \min_G V(G) &= \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{w}}\sim p_{\text{data}}(\tilde{\mathbf{w}})}[(D(G(\mathbf{z},\tilde{\mathbf{w}}),\tilde{\mathbf{w}})-1)^2]+\\ &+\alpha\,\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z}),\tilde{\mathbf{w}}\sim p_{\text{data}},\mathbf{x}\sim p_{\text{data}}}[|\Phi(G(\mathbf{z},\tilde{\mathbf{w}}))-\Phi(\mathbf{x})|],\end{aligned}$$

where $\tilde{\mathbf{w}}\in\mathbb{R}^T$ is the whispered utterance; $\mathbf{x}\in\mathbb{R}^T$ is the natural speech; $\mathbf{x}_r\in\mathbb{R}^T$ is a randomly chosen natural chunk within the batch; $G(\mathbf{z},\tilde{\mathbf{w}})\in\mathbb{R}^T$ is the enhanced speech; $D(\mathbf{x},\tilde{\mathbf{w}})$, $D(G(\mathbf{z},\tilde{\mathbf{w}}),\tilde{\mathbf{w}})$, and $D(\mathbf{x},\mathbf{x}_r)$ are the discriminator decisions for each input pair; and $\Phi(\mathbf{x})$ corresponds to the short-time Fourier transform magnitude in dBs (20 ms windows, 10 ms stride, and 2048 bias), with $\alpha=10^{-3}$ corresponding to the weighting of this term.

### *7.2. Results*

First of all, we perform an objective evaluation with mel cepstral distortion (MCD). MCD is an indicator of correct uttered content generation and speaker identity match in speech synthesis. We use the same formulation as in previous speech synthesis works (Pascual and Bonafonte, 2016; Pascual, 2016). Table 4 shows the results for the baseline RNN, the SEAE+ and WSEGAN. Firstly, we can see that SEAE+ has the highest distortion rate, indicating its lack of reconstruction capacity from the whispered signal towards the clean one. Actually, qualitative listenings allow us to appreciate how it is not able to reconstruct voiced segments, and the best it does is a low pass reconstruction of the input whispered signal itself. The application of the adversarial component is thus more critical in this setup as our intuition from Sec. 6.2.1 suggested. The qualitative listenings for WSEGAN reveal pitch reconstructions that match natural intonations and the expected modelled male identity (no low-pass effect is observed in this case). Regarding the RNN, it obtains the best score objectively, thus indicating possibly the best match to the clean signal. It is expectable to obtain such score as the model was directly optimized to minimize its quadratic error towards the clean spectral components. Nevertheless, low distortion scores are not always

Table 4: Mel cepstral distortion results for the three considered systems: RNN baseline, SEAE+ (L1 auto-encoder), and WSEGAN.

| | RNN | SEAE+ | WSEGAN |
|---|---|---|---|
| MCD [dB] | 8.01 | 17.19 | 12.81 |

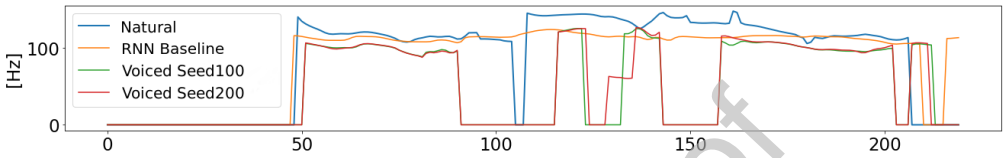


Figure 5: Natural pitch contour in blue and three example reconstructions. Orange is the RNN baseline contour, with a relatively flat behavior. Green and red are two different voiced versions from WSEGAN, produced with different latent codes $\mathbf{z}_1$ and $\mathbf{z}_2$ (i.e., different random seeds).

an indicator of a natural sounding voice in speech synthesis. In fact, a model with increased variance in its acoustic predictions (as in the case of WSEGAN), which in turn increments speech naturalness, can be an objectively inferior model (Henter et al., 2018). Hence, a subjective evaluation is normally the best procedure to assess the generated speech naturalness.

For the case of generated intonations of the two successful models, Fig. 5 lets us appreciate examples of generated pitch contours. We can first observe an increased variance in the pitch contours of the signal for WSEGAN as opposed to the RNN. The figure also shows different trajectories that match plausible intonation contours depending on a randomly selected latent description $\mathbf{z}_i$, which is enabled by the generative capacity of the model. We can also appreciate that the regenerated signal has different voiced/unvoiced regions (unvoiced regions are denoted by a 0 Hz signal). We hypothesize that these mismatches with the ground-truth signal may be corrected with the addition of more data, as the network could better estimate the right placement of pitch contours within the spoken contents of the damaged signal.

A subjective test was carried out to assess the improvement of WSEGAN with respect to the pitch regression RNN baseline system (online samples are

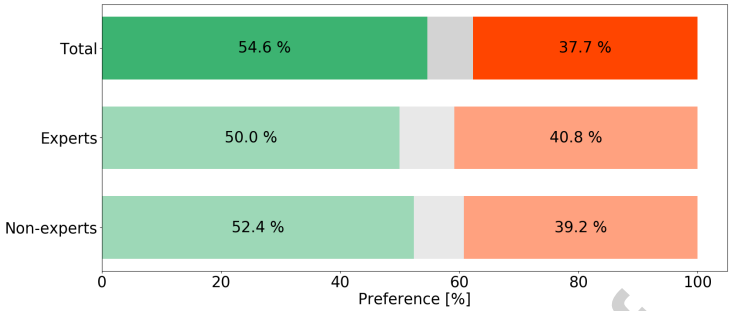


Figure 6: Subjective test preference results on naturalness rating between RNN regression baseline and WSEGAN. Green denotes WSEGAN preference, red denotes RNN preference and gray denotes that both are equally preferred.

referenced in Sec. 1 so that the reader can evaluate the differences qualitatively). A set of 25 subjects listened to and rated 10 randomly selected test utterances from a pool of 44, choosing whether they preferred the naturalness of one system, the other one, or both of them (the order of the two systems per utterance was shuffled). The results of this test are shown in Fig. 6, where WSEGAN (green) is preferred in 54.6% of the utterances, against the 37.7% of preference for the RNN system (red). Additionally, we observe no clear difference of preference between expert and nonexpert listeners (12 participants declared having expertise with speech signals and audio processing techniques). Participants noted that WSEGAN could sound more natural, implicitly producing proper intonations matching the sentences, but loses intelligibility in some utterances, potentially owing to the lack of data for such a large model. Interestingly, a native English listener even unveiled the geographic accent of the English speaker accent after WSEGAN recovery.

## 8. Conclusion

In this work, we propose a speech enhancement method framed within the GAN methodology using raw audio. We explore some variations of it that make it more efficient and effective. The model is an encoder-decoder fully convolutional structure, which makes it adaptable to deal with sequences of any length. With the introduced variations, we unveil some possible future paths to further improve the architecture, specifically in terms of encoder structure to obtain a better decimation scheme. The current results suggest

that our approach performs better than classic baselines such as Wiener or LogMMSE. They also show that the approach is competitive with custom-tuned deep learning models in the log-power spectral domain, trained with a regression on the magnitude, where a major amount of noise is detected and removed. Our approach, on the other hand, requires little preprocessing, working on the raw waveform, and is more flexible to work with other enhancement tasks such as speech reconstruction or whispered-to-voiced conversion. We also verify the effectiveness of the adversarial component over the fully convolutional regression system. This component becomes specially relevant in damaged signal reconstructions as in the WSEGAN setup, where plausible intonations and prosody are constructed matching the spoken contents.

## Acknowledgments

The authors thank José Andrés González for sharing the whispered data with them and providing constructive feedback in the corresponding experimentation. The authors also deeply thank the participants of the subjective evaluation. This work was supported by the Spanish Ministerio de Economía y Competitividad and European Regional Development Fund, contract TEC2015-69266-P (MINECO/FEDER, UE).

## References

, 2007. P.862.2: Wideband extension to recommendation p.862 for the assessment of wideband telephone networks and speech codecs.

Berouti, M., Schwartz, R., Makhoul, J., 1979. Enhancement of speech corrupted by acoustic noise. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Vol. 4. pp. 208–211.

Dendrinos, M., Bakamidis, S., Carayannis, G., 1991. Speech enhancement from noise: A regenerative approach. Speech Communication 10 (1), 45–57.

Donahue, C., Li, B., Prabhavalkar, R., 2018a. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 5024–5028.

Donahue, C., McAuley, J., Puckette, M., 2018b. Synthesizing audio with generative adversarial networks. ArXiv: 1802.04208.

Ephraim, Y., 1992. Statistical-model-based speech enhancement systems. Proceedings of the IEEE 80 (10), 1526–1555.

Ephraim, Y., Van Trees, H. L., 1995. A signal subspace approach for speech enhancement. IEEE Trans. on Speech and Audio Processing 3 (4), 251–266.

Erdogan, H., Hershey, J. R., Watanabe, S., Le Roux, J., 2015. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 708–712.

Fagan, M. J., Ell, S. R., Gilbert, J. M., Sarrazin, E., Chapman, P. M., 2008. Development of a (silent) speech recognition system for patients following laryngectomy. Medical Engineering & Physics 30 (4), 419–425.

Fu, S.-W., Tsao, Y., Lu, X., Kawai, H., 2017. Raw waveform-based speech enhancement by fully convolutional networks. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017. pp. 006–012.

Fu, S.-W., Wang, T.-W., Tsao, Y., Lu, X., Kawai, H., 2018. End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks. IEEE/ACM Transactions on Audio, Speech and Language Processing 26 (9), 1570–1584.

Gong, Y., Poellabauer, C., 2018. Impact of aliasing on deep cnn-based end-to-end acoustic models. Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH), 2698–2702.

Gonzalez, J. A., Cheah, L. A., Gomez, A. M., Green, P. D., Gilbert, J. M., Ell, S. R., Moore, R. K., Holdsworth, E., 2017a. Direct speech reconstruction from articulatory sensor data by machine learning. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25 (12), 2362–2374.

Gonzalez, J. A., Cheah, L. A., Green, P. D., Gilbert, J. M., Ell, S. R., Moore, R. K., Holdsworth, E., 2017b. Evaluation of a silent speech interface based

on magnetic sensing and deep learning for a phonetically rich vocabulary. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 3986–3990.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C., 2017. Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 5767–5777.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV). pp. 1026–1034.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 770–778.

Henter, G. E., King, S., Merritt, T., Degottex, G., 2018. Analysing shortcomings of statistical parametric speech synthesis. arXiv preprint arXiv:1807.10941.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6626–6637.

Higuchi, T., Kinoshita, K., Delcroix, M., Nakatani, T., 2017. Adversarial training for data-driven speech enhancement without parallel corpus. In: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). pp. 40–47.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735–1780.

Hu, Y., Loizou, P. C., 2008. Evaluation of objective quality measures for speech enhancement. IEEE Trans. on Audio, Speech, and Language Processing 16 (1), 229–238.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. of the Int. Conf. on Machine Learning (ICML). pp. 448–456.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., 2017. Image-to-image translation with conditional adversarial networks. In: Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 5967–5976.

Kingma, D. P., Ba, J. L., 2015. Adam: a method for stochastic optimization. In: Proc. of the Int. Conf. on Learning Representations (ICLR).

Kominek, J., Black, A. W., 2004. The CMU Arctic speech databases. In: Fifth ISCA Workshop on Speech Synthesis. pp. 223–224.

Lim, J., Oppenheim, A., 1978. All-pole modeling of degraded speech. IEEE Trans. on Acoustics, Speech, and Signal Processing 26 (3), 197–210.

Loizou, P. C., 2013. Speech Enhancement: Theory and Practice, 2nd Edition. CRC Press, Inc., Boca Raton, FL, USA.

Lu, X., Tsao, Y., Matsuda, S., Hori, C., 2013. Speech enhancement based on deep denoising autoencoder. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 436–440.

Maas, A. L., Le, Q. V., O'Neil, T. M., Vinyals, O., Nguyen, P., Ng, A. Y., 2012. Recurrent neural networks for noise reduction in robust ASR. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 22–25.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., Smolley, S. P., 2017. Least Squares Generative Adversarial Networks. In: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV). IEEE, pp. 2813–2821.

Meng, Z., Li, J., Gong, Y., et al., 2018. Cycle-consistent speech enhancement. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH).

Nakamura, K., Janke, M., Wand, M., Schultz, T., 2011. Estimation of fundamental frequency from surface electromyographic data: Emg-to-f 0. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 573–576.

Nakamura, K., Toda, T., Saruwatari, H., Shikano, K., 2012. Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech. Speech Communication 54 (1), 134–146.

Narayanan, A., Wang, D., 2013. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 7092–7096.

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill 1 (10), e3.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al., 2017. Parallel wavenet: Fast high-fidelity speech synthesis. A rXiv:1711.10433.

Ortega-Garcia, J., Gonzalez-Rodriguez, J., 1996. Overview of speech enhancement techniques for automatic speaker recognition. In: Proc. of the Int. Conf. on Spoken Language Processing (ICSLP). Vol. 2. pp. 929–932.

Paliwal, K., Wójcicki, K., Shannon, B., 2011. The importance of phase in speech enhancement. Speech Communication 53 (4), 465 – 494.

Park, S. R., Lee, J., 2017. A fully convolutional neural network for speech enhancement. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH).

Parveen, S., Green, P., 2004. Speech enhancement with missing data techniques using recurrent neural networks. In: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 733–736.

Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318.

Pascual, S., 2016. Deep learning applied to speech synthesis. Master's thesis, Universitat Politècnica de Catalunya.

Pascual, S., Bonafonte, A., 2016. Multi-output RNN-LSTM for multiple speaker speech synthesis and adaptation. In: Proc. 24th European Signal Processing Conference (EUSIPCO). IEEE, pp. 2325–2329.

Pascual, S., Bonafonte, A., Serrà, J., 2017. Segan: Speech enhancement generative adversarial network. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 3642–3646.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. In: NeurIPS Workshop on The Future of Gradient-based Machine Learning Software & Techniques (NeurIPS-Autodiff).

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A., 2016. Context encoders: Feature learning by inpainting. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2536–2544.

Qin, S., Jiang, T., 2018. Improved wasserstein conditional generative adversarial network speech enhancement. EURASIP Journal on Wireless Communications and Networking 2018 (1), 181.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv: 1511.06434.

Rethage, D., Pons, J., Serra, X., 2018. A wavenet for speech denoising. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 5069–5073.

Scalart, P., Filho, J. V., 1996. Speech enhancement based on a priori signal to noise estimation. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Vol. 2. pp. 629–632 vol. 2.

Shivakumar, P. G., Georgiou, P. G., 2016. Perception optimized deep denoising autoencoders for speech enhancement. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 3743–3747.

Taal, C. H., Hendriks, R. C., Heusdens, R., Jensen, J., 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 4214–4217.

Taal, C. H., Hendriks, R. C., Heusdens, R., Jensen, J., 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. IEEE

Transactions on Audio, Speech, and Language Processing 19 (7), 2125–2136.

Tamura, S., Waibel, A., 1988. Noise reduction using connectionist models. In: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 553–556.

Thiemann, J., Ito, N., Vincent, E., 2013. The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings. Journal of the Acoustical Society of America 133 (5), 3591–3591.

Tieleman, T., Hinton, G., 2012. Lecture 6.5-RMSprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning 4, 2.

Toda, T., Black, A. W., Tokuda, K., 2008. Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model. Speech Communication 50 (3), 215–227.

Valentini-Botinhao, C., Wang, X., Takaki, S., Yamagishi, J., 2016. Investigating rnn-based speech enhancement methods for noise-robust text-to-speech. In: 9th ISCA Speech Synthesis Workshop. pp. 146–152.

Veaux, C., Yamagishi, J., MacDonald, K., et al., 2016. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

Wang, D., Lim, J., 1982. The unimportance of phase in speech enhancement. IEEE Trans. on Acoustics, Speech, and Signal Processing 30 (4), 679–681.

Wang, Y., Narayanan, A., Wang, D., 2014. On training targets for supervised speech separation. IEEE/ACM Transactions on Audio, Speech and Language Processing 22 (12), 1849–1858.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., Schuller, B., 2015. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In: Proc. of the Int. Conf. on Latent Variable Analysis and Signal Separation. pp. 91–99.

Weninger, F., Hershey, J. R., Le Roux, J., Schuller, B., 2014. Discriminatively trained recurrent neural networks for single-channel speech separation. In:

Proc. of the IEEE Global Conf. on Signal and Information Processing (GlobalSIP)A.

Williamson, D. S., Wang, D., 2017. Time-frequency masking in the complex domain for speech dereverberation and denoising. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25 (7), 1492–1501.

Xia, B., Bao, C., 2013. Speech enhancement with weighted denoising auto-encoder. In: Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH). pp. 3444–3448.

Xu, Y., Du, J., Dai, L.-R., Lee, C.-H., 2015. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Transactions on Audio, Speech and Language Processing 23 (1), 7–19.

Yang, L.-P., Fu, Q.-J., 2005. Spectral subtraction-based speech enhancement for cochlear implant patients in background noise. Journal of the Acoustical Society of America 117 (3), 1001–1004.

Yu, D., Deng, L., Droppo, J., Wu, J., Gong, Y., Acero, A., 2008. A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 4041–4044.

Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2018. Self-attention generative adversarial networks. ArXiv: 1805.08318.

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: