Adversarial Training for Speech Super-Resolution

Sefik Emre Eskimez, *Student Member, IEEE,* Kazuhito Koishida, *Member, IEEE* Zhiyao Duan, *Member, IEEE*

***Abstract*—Speech super-resolution or speech bandwidth expansion aims to upsample a given speech signal by generating the missing high-frequency content. In this paper, we propose a deep neural network approach exploiting the adversarial training ideas that have been shown effective in image super-resolution. Specifically, our proposed network follows the Generative Adversarial Networks (GAN) setup, where the generator network uses a convolutional autoencoder architecture with 1D convolution kernels to generate high-frequency log-power spectra from the low-frequency log-power spectra of the input speech. We propose to use both the reconstruction loss and the adversarial loss for training, and we employ a recent regularization method that penalizes the gradient norms of the discriminator to stabilize the training. We compare our proposed approach with two state-of-the-art neural network baselines and evaluate these methods with both objective speech quality measures and subjective perceptual and intelligibility tests. Results show that our proposed method outperforms both baselines in terms of both objective and subjective evaluations. To gain insights of the network architecture, we analyze key parameters of the proposed network including the number of layers, the number of convolution kernels, and the relative weight of the reconstruction and adversarial losses. Besides, we analyze the computational complexity of our method and the baselines and discuss ways for phase estimation. We further develop a noise-resilient version of the proposed approach by training the network with noisy speech inputs. Objective evaluation validates the noise-resilient property on unseen noise types.**

***Index Terms*—speech super-resolution, artificial bandwidth expansion, generative adversarial networks, 1D convolutional neural networks, speech processing**

I. Introduction

Deep neural networks (DNNs) have been outperforming traditional methods in various classification and regression tasks, and speech processing is not an exception. For speech recognition, enhancement, emotion recognition, and speaker identification/verification, state-of-the-art methods are based on DNNs.

An interesting problem in speech processing is to expand the bandwidth of speech signals by generating the missing high frequencies (i.e., increasing the waveform resolution). This problem is named *artificial speech bandwidth expansion* or Speech Super-Resolution (SSR) in the literature. In this paper, we tackle this problem and refer it SSR.

SSR is beneficial for speech communication over low-bandwidth channels. An SSR module can be integrated into receiver-end devices to enhance the resolution of transmitted low-resolution signals. One study shows that users prefer a wider frequency range in communication [1]. Other studies show that the narrowband communication is challenging for the hearing impaired population [2], and artificially expanding the bandwidth up to 8 kHz leads to improved speech recognition rates for Cochlear Implant (CI) users [3]. Furthermore, speech synthesis systems can also benefit from employing a computationally light-weight SSR module after synthesizing low-resolution speech. This is because the computational cost of speech synthesis drastically increases as the sampling rate increases, preventing a real-time high-resolution synthesis on edge computing devices. Also, speech synthesis systems, once trained, are not straightforward to change the sampling rate on the fly.

In this paper, we propose a novel neural network framework that leverages adversarial training for SSR, and utilize a recent regularization method that stabilizes the adversarial training. We employ a sequence-to-sequence convolutional autoencoder network that accepts Log Power Spectrogram (LPS) as input and yields the corresponding high-frequency range LPS. We use 1D kernels in the convolutional layers that operate along the time axis of the spectrogram. The training process contains two major steps. First, we train our network using only a reconstruction loss for a few epochs as the initialization. Then, we switch to the adversarial loss in addition to the weighted reconstruction loss.

We train our network on the Centre for Speech Technology Research (CSTR) Voice Cloning Toolkit (VCTK) Corpus [4] and evaluate it on an entirely disjoint dataset to show the robustness against unseen speakers and recording conditions, namely the Wall Street Journal (WSJ0) corpus [5]. We compare with [6], [7] baselines. The objective and subjective evaluations show that the resulting enhanced time domain signals yield better results than the baseline methods. We further analyze our network by changing the network parameters, namely the number of layers and filters in the autoencoder, and the reconstruction loss weight parameter, and report the objective scores. Besides, we discuss the stability of GAN training for different regularization methods and compare phase estimation methods. Furthermore, we compare the computational complexity of our method and the baselines. We also propose a method to train the network against the noise, and we analyze it against the unseen non-stationary noise types. In addition, we conducted a listening test to verify the intelligibility of the generated samples. Some examples of synthesized super-resolution speech are publicly available[1].

In summary, our contributions in this work are as follows:

- We apply the generative adversarial network framework to speech super-resolution and synthesize the high-

S. E. Eskimez and Z. Duan are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, 14627, USA. E-mail: eeskimez@ur.rochester.edu, zhiyao.duan@rochester.edu.

K. Koishida is with the Microsoft Corporation, One Microsoft Way, Redmond, WA, USA. E-mail: kazukoi@microsoft.com.

The work was mainly done when S. E. Eskimez was an intern at Microsoft Research. The work was also partially supported by the National Science Foundation grant No. 1617107.

[1] http://www.ece.rochester.edu/projects/air/projects/SSRGAN.html

resolution speech spectrogram directly with the network.

- We use a regularization method [8] to address the failure modes encountered during GAN training, and effectively stabilize it.
- We obtain a computationally light-weight generator compared to the baselines due to the usage of 1D kernels in the convolutional layers.

The rest of the paper is organized as follows: in Section II, we describe the existing works on audio and speech super-resolution, and outline Generative Adversarial Networks (GANs). Section III describes the system overview, the neural network architectures, and the loss functions. In Section IV, we describe the datasets employed for training and evaluation, and how we prepared the data for training and inference, we describe the objective and subjective evaluation methods and results, and we analyze the network. We conclude the paper in Section V.

II. Related Work

A. Artificial Speech Bandwidth Expansion

Speech Super-Resolution (SSR) is studied widely by the research community under the name of *artificial speech bandwidth expansion* [9], [10], [11], [6]. In [9], Park et al. used Linear Predictive Coding (LPC) coefficients, pitch, and power that were extracted from the narrowband signal, and modeled the mapping between narrowband and wideband parameters using a Gaussian Mixture Model (GMM). Chennoukh et al. [12] proposed a method that extends the bandwidth using Line Spectral Frequencies (LFS), applied on LPC coefficients. Seo et al. [11] proposed a GMM model for maximum a posterior estimation of the wideband spectrum from the narrowband. This method also considers sentence-level temporal dynamics to synthesize wideband speech. Jax et al. [13] proposed a method to estimate the gain and the shape of the spectral envelope of the wideband using a Hidden Markov Model (HMM). Song et al. [14] showed that the Baum-Welch re-estimation algorithm outperforms the method proposed by Jax et al. [13]. They also showed that the GMM-based methods are a special case of the HMM-based methods, while their performances are comparable. Abel et al. [15] proposed to use DNNs for high band spectral envelope estimation, and compared with GMM and HMM-based baselines. They showed that DNNs outperform the baselines.

While some works focused on predicting the wide-band spectral envelope, others focused on directly estimating the missing data points [16], [17], [6], [7]. In [16], the authors used a latent component analysis and Expectation-Maximization (EM) algorithm to estimate missing frequencies, similar to Non-negative Matrix Factorization (NMF). Sun et al. [17] cast the bandwidth extension problem as a convex optimization problem and employed NMF to estimate the missing frequencies. In one of the notable works, Li et al. [6] proposed a DNN to predict the log-power spectrum of the wideband. They used 32 ms window size and 16 ms hop size when extracting LPS features from the input narrowband. The hidden layers were pre-trained using the Restricted Boltzmann Machine (RBM). Their network accepts nine frames of narrowband LPS and predicts a single frame of wideband LPS. Since phase information is still missing, they flip the phase of the low-frequency band as that of the high-frequency band to reconstruct the time domain signal. They trained and evaluated their method on the Wall Street Journal (WSJ0) Corpus. They showed that their method yields better results compared to the GMM baseline in both objective and subjective evaluations.

Kuleshov et al. [7] proposed an end-to-end super-resolution method that takes the raw waveform as input and outputs the super-resolution waveform. They employed 1D convolution layers and formed an auto-encoder with concatenating skip connections, which are similar to skip connections but instead of adding the feature maps together, they are concatenated. Before being fed to the network, the low-resolution waveform is upsampled to match the sampling rate of the target super-resolution signal. This upsampled input is also added to the network output. A Mean-Squared Error (MSE) loss function is used for training. Compared with neural methods working with time-frequency representations, one significant advantage of this time domain approach is that no special module is needed to estimate the signals' phase. However, it is computationally very expensive.

B. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [18] have been employed to generate highly realistic images, videos and speech signals. In essence, GANs contain two neural networks, a generator, and a discriminator. The generator tries to generate fake but realistic data, while the discriminator tries to distinguish between the real and fake data. When the training converges, the generator is able to generate data that lie on the real data manifold, and the discriminator cannot tell the fake from real data. There are variants of GANs, which improve the generation capability or add controls over the generated distributions. Deep Convolutional GAN (DCGAN) [19] can generate realistic images, where both the generator and discriminator architectures are based on convolutional neural networks. The conditional GANs [20] are another family of GANs where the generator and discriminator accepts a condition input and enables control over the generated distribution.

Although GANs are powerful, they suffer from instabilities during training [21], which lead GANs not to converge and make them yield poor results. Therefore, researchers steered towards finding better training methods for GANs [22], [23], [24], [8], [21]. Wasserstein GAN (WGAN) [22] is one of the regularized GAN family members that employs the Wasserstein divergence instead of the Jensen-Shannon divergence and maintains the Lipschitz constraint by clipping the weights. In an improved version of WGAN [23], instead of weight-clipping, Gulrajani et al. proposed a Gradient-Penalty (GP) to satisfy the Lipschitz constraint. In the proposed method, the data point between a real and generated distributions is drawn, and the norm of the gradient for this data point is penalized for not having a unit norm. For WGAN and WGAN-GP, the critic (discriminator) is usually updated for a few iterations before alternating to updating the generator, which makes the training computationally intense. Another regularization technique is to add instance noise, which is typically chosen as Gaussian noise, to the input of the discriminator [24]. Mescheder et al. [21] show that instance noise is indeed useful for GAN training, and leads GANs to converge. Roth et al. [8] derived a zero-centered GP regularizer that is inspired from the instance noise. Mescheder et al. [21] proposed two similar but simplified versions of Roth et al.'s regularizer, one of them only penalizes the generated data distribution, while the other one only penalizes the real data distribution. In this work, we choose to penalize both the real and generated distribution; therefore we use the regularizer proposed by Roth et al. [8].

GANs have been successfully applied to image and video super-resolution. Ledig et al. [25] confirmed that reconstruction loss based single image super-resolution systems yield blurry results. By using an adversarial training loss, they showed that their Super-Resolution Generative Adversarial Network (SRGAN) yields sharper, superior results that lie on the data manifold. GANs also benefit Video Super-Resolution (VSR). Lucas et al. [26] showed that their GAN based VSR system outperforms the current state-of-the-art VSR systems. These studies inspired us to investigate the application of GANs to SSR, where we work with spectrograms that are similar to images or video frames.

It is noted that Li et al. [27] has proposed a GAN-based SSR approach recently. They employed a fully connected neural network (generator) with two hidden layers to predict the Line Spectral Frequencies (LSF) and speech energy of the high band (HB) from LSF, delta LSF and speech energy of the low band signal. They used a fully connected discriminator to distinguish fake parameters from real parameters. They then used the EVRC-WB framework [28] and a synthesis filterbank to synthesis high-resolution speech signals from the predicted speech parameters. Although our approach is similar to [27] in the sense that they are both applications of GANs in SSR, one of the key differences is that we directly generate the speech spectrograms, while [27] generates speech parameters (LSF + energy) and synthesize speech from those parameters with another synthesis framework. Another novelty of our work is that we use a recently proposed regularizer [8] to stabilize GAN training. Furthermore, our generator and discriminator architectures contain convolutional layers, while [27] uses only fully connected layers.

III. Proposed SSR System

A. System Overview

We propose a neural network approach with adversarial training to tackle the Speech Super-Resolution (SSR) problem. Before we introduce the network architecture and training processes, we think it is helpful to first explain how the whole SSR system runs during test time, treating the network as a black box. This process is shown in Figure 1. Let x be the time domain waveform of the narrowband speech that we want to increase the time resolution. First, the Short-Time Fourier Transform (STFT) is applied to x with parameter settings described in Section IV-D. The Log-Power Spectrogram (LPS) X^{NB} and the phase spectrogram X_P are computed from X, and X^{NB} is fed to the proposed generator network, or namely the Speech-Super Resolution Generative Adversarial Network (*SSR-GAN*) to estimate the high-frequency range LPS, $\hat{X}^{WB}$. The original narrowband and the predicted high-frequency range are concatenated to obtain the estimated wideband LPS X^{SR}. In order to avoid discontinuities at the concatenation [6], we also predict the highest C frequency bins of the narrowband spectrogram, where C is called the *offset* parameter. During concatenation, the top C frequency bins are removed from the narrowband spectrogram.

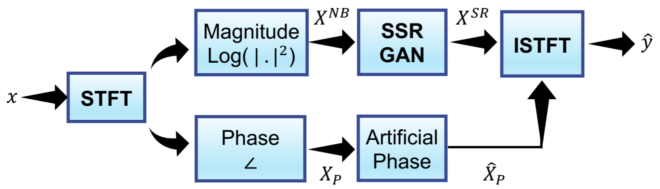


Fig. 1: Overview of the proposed SSR system during test time. The Log-Power Spectra (LPS) X^{NB} and the phase spectrogram X_P are calculated from the input narrowband waveform x through Short-Time Fourier Transform (STFT). X^{NB} is fed to the speech super-resolution generative adversarial network (*SSR-GAN*) to obtain the estimated high-frequency range LPS $\hat{X}^{WB}$, which is then concatenated with the original narrowband LPS. The phase of the high-frequency range is artificially produced by flipping and repeating the narrowband phase X_P and adding a negative sign. For fractional super-resolution factors, the last repeat is truncated to match the frequency range. Finally, the estimated wideband LPS and artificial phase are used to reconstruct the time-domain signal $\hat{y}$ by Inverse STFT (ISTFT) and overlap-add.

Since we do not estimate the phase of the high frequencies, we follow Li et al. [6] to create an artificial phase by flipping the narrowband phase and reverting the sign. For the 2x super-resolution version, we concatenate this flipped phase with the narrowband phase to obtain an artificial phase $\hat{X}_P$ of the entire wideband signal. For the 4x super-resolution version, we repeat the flipped phase three times. For fractional super-resolution factors, the last repeat is truncated to match the frequency range. Our method could be improved by predicting the phase from the magnitude spectrogram; however, this is a challenging problem itself [29].

Finally, inverse STFT is applied to the complex spectrogram calculated from the estimated wideband LPS X^{SR} and artificial phase $\hat{X}_P$, and the time domain signal $\hat{y}$ is reconstructed using the overlap-add method.

B. Network Architecture

In this section, we explain the generator and discriminator architectures. The generator is fully convolutional, while the discriminator contains convolutional layers followed by two Fully Connected (FC) layers. The architectures are shown in Figure 2.

For the generator network, we employ a common bottleneck autoencoder architecture described in [7]. The generator is a

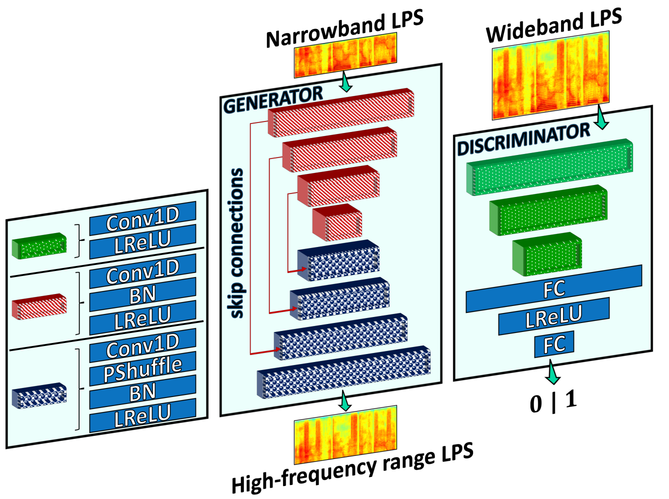


Fig. 2: The proposed network architectures for the generator (middle) and the discriminator (right). Each rectangular block is a convolutional layer with structures color coded and detailed on the left subfigure. The generator is an autoencoder with concatenating skip connections, predicting the high-frequency range of the input narrowband magnitude spectrogram. It is then concatenated with the original low-frequency range to generate the full wideband magnitude spectrogram. The input to the discriminator is the full wideband spectrogram of either a real sample or a generated sample. We do not use batch normalization in the discriminator. Notations: *BN* - batch normalization layer, *FC* - fully connected layer, *LReLU* - LeakyReLU activation, and *PShuffle* - pixel shuffle or sub-pixel layer, *LPS* - log-power spectrogram.

sequence-to-sequence model that accepts the narrowband LPS with T time steps and outputs the high-frequency range LPS with T time steps.

In the generator network, we use a Batch Normalization (BN) layer after each convolutional layer and before the activation. BN allows the network to converge faster and allows higher learning rates to be used for training. We use sub-pixel (or pixel shuffle) layers introduced in [30], which is proved useful for image and video super-resolution. The main idea behind the sub-pixel layers is to compute more feature maps on the convolution layer and resize them into an upsampled data. Readers are referred to see [30] for more details about sub-pixel layers. We use leaky rectified linear units (LeakyReLU) as the activation with a slope of 0.2, except for the output layer, where we use linear activation.

The discriminator network accepts the concatenated narrowband and high-frequency range LPSs as input, where the high-frequency range LPS could be generated by the generator network or coming directly from the data distribution. Including the narrowband to the discriminator's input is essentially conditioning the input high-frequency range LPS on the narrowband LPS, similar to conditional GANs [31]. The discriminator contains three convolutional layers as shown in Figure 2. Different from the generator, we do not employ BN layers in the discriminator. Using BN in the discriminator leads to instabilities during training, especially if the discriminator loss is regularized [8], [21]. The convolutional layers are followed by two FC layers. We use LeakyReLU activation with a slope of 0.2 in all layers, except for the output layer, where we use a linear activation function. The details of both network architectures are shown in Table I.

C. Loss Functions

In this section, we describe the training objectives of the generator and the discriminator. First, we train our network using a reconstruction loss as initialization for several epochs. This process lets the generator to produce the "mean" results, which are overly smooth. Then, we switch to using both the reconstruction loss and an adversarial loss (GAN loss). Using GAN loss produces sharper and more detailed LPSs. We use a parameter to weight these two losses in the generator's objective function. In the following, we explain the details for each loss function.

1) Reconstruction Loss: There are a few candidates for the reconstruction loss. The most common distance functions are L1-norm and L2-norm, or namely, Mean Absolute Error (MAE) and Mean Squared Error (MSE). Our initial testing showed that using Log-Spectral Distance (LSD) (or Log-Spectral Distortion) function as our training objective yield slightly better results for the SSR task. The LSD measures the distance between two spectrograms in decibels, and it is mathematically defined as follows:

$$l_{LSD} = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{1}{K}\sum_{k=1}^{K}[X^{HR}(l,k) - X^{SR}(l,k)]^2}, \qquad (1)$$

where X^{HR} and X^{SR} are the ground truth and estimated LPS, respectively K is the number of frequency bins. LSD is used widely for evaluating SSR methods objectively. In this work, we use it as both the reconstruction loss and an objective evaluation metric. LSD is essentially the average L2 distance of LPS across time frames.

2) Adversarial Loss: The original generative adversarial network (GAN) is a two player, zero-sum (minimax) game between a generator and a discriminator. The generator's job is to generate realistic data that can fool the discriminator into classifying it as real data, while the discriminator's job is to distinguish the real and fake data apart. When this game reaches a Nash equilibrium, the generator is able to produce realistic data that the discriminator cannot tell from real data. In the SSR context in this paper, this two-player game can be defined as follows:

$$\begin{gathered}\min_{\theta}\max_{\psi}\mathbb{E}_{\mathbb{P}}[\log D_{\psi}(X^{HR})] + \mathbb{E}_{\mathbb{Q}}[\log(1 - D_{\psi}(G_{\theta}(X^{NB})))], \\ \mathbb{P} : X^{HR} \sim p(X^{HR}) \\ \mathbb{Q} : X^{NB} \sim p(X^{NB})\end{gathered} \qquad (2)$$

where X^{HR} is the high resolution data (real data), X^{NB} is the narrowband data. $G_{\theta}(\cdot)$ is the generator and $D_{\psi}(\cdot)$ is the discriminator, where θ and ψ represent their trainable parameters. $\mathbb{P}$ is the distribution of real data and $\mathbb{Q}$ is the distribution of the narrowband data. This formulation assumes

TABLE I: Detailed parameters of the proposed network architecture. The number of channels and hidden units, filter sizes, strides, activations and output shapes are shown for each layer in the generator and discriminator networks. K and N are the narrowband and the high-frequency range LPS dimensions along the frequency axis, respectively. K is 129 and 65 for 2x and 4x super-resolution scales, respectively. N is 141 and 199 for 2x and 4x super-resolution scales, respectively.

Net	Layer	Activation	Filter No.	Filter Size	Stride	BN	Sub-Pix	Output Shape
Generator	Input	-	-	-	-	-	-	$32 \times K$
	Conv	LeakyReLU	256	(7, 1)	(2, 1)	Yes	No	16×256
	Conv	LeakyReLU	512	(5, 1)	(2, 1)	Yes	No	8×512
	Conv	LeakyReLU	512	(3, 1)	(2, 1)	Yes	No	4×512
	Conv	LeakyReLU	1024	(3, 1)	(2, 1)	Yes	No	2×1024
	Conv	LeakyReLU	512	(3, 1)	(1, 1)	Yes	Yes	4×512
	Conv	LeakyReLU	512	(5, 1)	(1, 1)	Yes	Yes	8×512
	Conv	LeakyReLU	256	(7, 1)	(1, 1)	Yes	Yes	16×256
	Conv	LeakyReLU	N	(7, 1)	(1, 1)	Yes	Yes	$32 \times N$
	Conv	LeakyReLU	N	(9, 1)	(1, 1)	No	No	$32 \times N$
Discriminator	Input	-	-	-	-	-	-	$32 \times (K + N)$
	Conv	LeakyReLU	1024	(7, 1)	(2, 1)	No	No	16×1024
	Conv	LeakyReLU	1024	(5, 1)	(2, 1)	No	No	8×1024
	Conv	LeakyReLU	1024	(3, 1)	(2, 1)	No	No	4×1024
	Flatten							4096
	FC	LeakyReLU	2048			No		2048
	FC	Sigmoid	1			No		1

the generator contains the concatenation of narrowband LPS and high-frequency LPS. This equation can be simplified as follows:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{\mathbb{P}}[\log \varphi_R] + \mathbb{E}_{\mathbb{Q}}[\log(1 - \varphi_F)], \tag{3}$$

where φ_R and φ_F are the discriminator output for real and fake data, respectively.

In practice, unregularized GANs are usually unstable during training, depending on the task at hand, and do not always converge [21]. To stabilize the GAN training, we add a penalty on the weighted gradient-norms of the discriminator as described in [8]. The regularization term is described as:

$$\Omega = \mathbb{E}_{\mathbb{P}}[(1 - \varphi_R)^2 \|\nabla \phi_R\|^2] + \mathbb{E}_{\mathbb{Q}}[\varphi_F^2 \|\nabla \phi_F\|^2], \tag{4}$$

where $\phi = \sigma^{-1}(\varphi)$, and σ is the sigmoid activation used in generating the output of the discriminator. Note that the gradients are computed on ϕ, before the sigmoid activation, which yields more robust gradients [8]. We add this term into the traditional GAN loss and obtain the loss for the discriminator as follows:

$$l_{DIS} = \mathbb{E}_{\mathbb{P}}[\log \varphi_R] + \mathbb{E}_{\mathbb{Q}}[\log(1 - \varphi_F)] - \frac{\gamma}{2}\Omega, \tag{5}$$

where γ is the weight for the regularization term.

The generator loss is defined as the weighted sum of the reconstruction loss and the adversarial loss. We minimize the following function:

$$l_{GEN} = \mathbb{E}_{\mathbb{Q}}[-\log(D_\psi(G_\theta(X^{NB})))] + \lambda l_{LSD}, \tag{6}$$

where l_{LSD} is the loss function described in Equation (1) and λ is the weighting factor for the LSD loss.

IV. Experiments

In this section, first, we describe the data used in this study and how we prepared the data for network training. Next, we describe the objective metrics used for evaluating our method. Then, we show the results of our experiments and analyze our network architecture by changing parameters. Next, we investigate our network's resilience to background noise, propose a training method to make the network robust against noise. Finally, we conclude this section by describing and presenting the results of a subjective evaluation of our method.

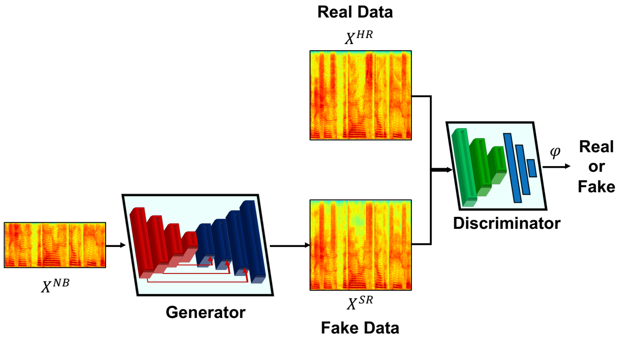


Fig. 3: The adversarial training procedure for the proposed method. The generator contains the concatenation of narrowband LPS and high-frequency LPS.

A. Datasets

The CSTR Voice Cloning Toolkit Corpus (VCTK), which is originally designed for training Text-to-Speech (TTS) synthesis systems, was used to train our network. There are a total of 109 English speakers with different accents. The recordings are 16-bit WAV files with 48 kHz sampling rate and contain clear speech. Each speaker utters 400 sentences, where the sentences are either taken from newspaper articles, the International Dialects of English Archive's Rainbow passages or an elicitation passage that aims to identify the speaker's accent. We split this dataset into training and validation sets, where we randomly chose six speakers for the validation set and use the rest for the training set.

We employed another dataset for evaluation that has different speakers and different recording conditions than the VCTK corpus, in order to evaluate the generalization capability of our network. This is the Wall Street Journal (WSJ0) corpus, where the speakers read the Wall Street news articles plus spontaneous dictations. The sampling rate of the recordings is 16 kHz. The recordings contain natural background noise. We randomly selected 5000 samples (around 12 hours) within this dataset for the objective evaluations.

We applied a low-pass filter and downsampled the high-resolution signals to obtain their parallel low-resolution signals for training and testing. We describe the details of this pre-processing applied to VCTK and WSJ0 datasets in Section IV-D.

B. Objective Metrics

To evaluate our method and compare it with the baselines, we employed LSD defined by Equation (1), Segmental Signal-to-Noise Ratio (SegSNR) [32], and Perceptual Evaluation of Speech Quality (PESQ) [33] objective metrics, which are widely used for SSR and speech enhancement literature. LSD measures the similarity between two spectrograms in decibels and defined in Equation (1), where a lower value is better. SegSNR is the signal-to-noise (SNR) ratio, averaged over segments of audio samples. It is defined as:

$$SegSNR = \frac{1}{L}\sum_{l=1}^{L} 10log\frac{\sum_{n=1}^{N}[x(l,n)]^2}{\sum_{n=1}^{N}[x(l,n)-\hat{x}(l,n)]^2}, \qquad (7)$$

where L is the number of segments, and N is the number of data points in the utterance. A higher value of SegSNR is better.

PESQ measures speech quality and it is standardized by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T). It is widely used in industry to assess the quality of telephony speech and in research fields such as speech enhancement. PESQ ranges from -0.5 to 4.5, where higher values correspond to better speech quality.

C. Baseline Methods

We chose two state-of-the-art methods described in Section II as comparison baselines. The first baseline is an FFT-based method [6], which we name as *BL1* through the rest of the paper. The neural network architecture of *BL1* is a DNN with three hidden layers with 2048 hidden units per hidden layer. The network accepts nine STFT frames, including four past and four future frames, and generates a single STFT frame. The objective function of this network is MSE. We implemented *BL1* as described in the original paper, except that we used VCTK corpus for training in order to fairly compare all methods. Since this work only considers 2x SSR, we did not implement 4x SSR version of this work.

The second baseline is a waveform-based method [7], which we name as *BL2* through the rest of the paper. Similar to ours, this network is a convolutional autoencoder, although our network is applied to spectrograms instead of waveforms. Another difference is that their network has an additive residual connection between the input and output of the network. The number of filters of the convolutional encoder layers is 128, 256, 512, and 512, and is 512 for the bottleneck layer. The decoder has twice the number of filters in the encoder layers but in reverse order. The size of filters of the convolutional encoder layers is 65, 33, 17, and 9, and is 9 for the bottleneck layer. The size of filters in the decoder layers are the same as the encoder but in reversed order. Their network is trained with the MSE objective function. For implementation, we used the code provided by the authors directly to generate results for both 2x and 4x SSR, using the hyperparameters described in their paper. To ensure fairness, we used the exact same data we used for our method during training and testing the baselines.

D. Pre-Processing

For our method, we applied the band-limited sinc interpolation method described in [34] to the high-resolution signal and obtained the downsampled signal. We computed the short-time Fourier transform (STFT) on both low and high-resolution signals, with 32 ms window size and 8 ms hop size. We applied the log and power operations to these spectrograms to obtain log-power spectra (LPS). We chopped up the utterances into T timesteps and form our dataset with narrowband and high-frequency range LPS pairs.

Similarly, for *BL1*, we followed the same steps. However, we followed their original implementation, and instead of 8 ms hop size, we used 16 ms hop size.

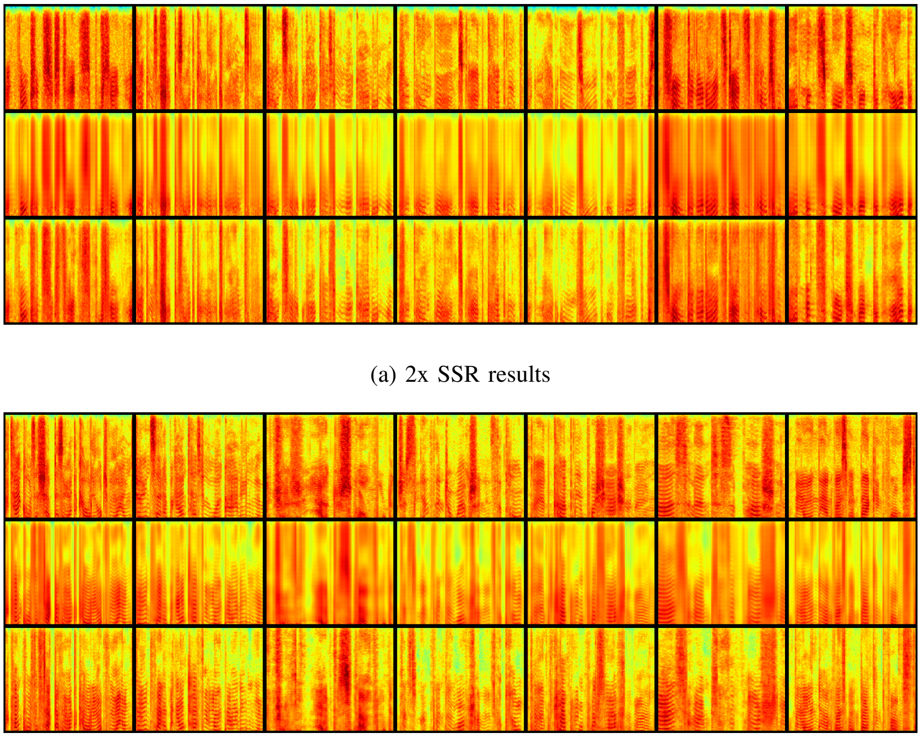
For the pre-processing for *BL2*, we used the author's code, which is available online. The low-resolution signals were created by applying an order 8 Chebyshev type I low-pass filter and downsampling the high-resolution signals. The low-resolution signals were upsampled to match the size of the high-resolution signals using cubic upscaling as the input to their neural network. The samples were chopped into patches with the length of 6000 in the high-resolution space (0.375 seconds), which is the same for 2x and 4x scales.

E. Implementation Details of Proposed Method

We implemented our system in Tensorflow [35]. We used mini-batches during training, and we set the mini-batch size to 64. We trained our network using only the LSD loss for 50 epochs, and then switched to LSD plus GAN loss for 100 epochs. We decided the number of epochs empirically. We still use LSD loss during GAN training, which keeps the output around the mean distribution as discussed in [25]. The number of time-steps T of our input and output spectrograms is 32. We used a learning rate of 10^{-4} when training the network using only LSD loss, and we used a learning rate of 10^{-5} for both the generator and discriminator when training the network using LSD plus GAN losses. We chose lower learning rate during GAN training to further stabilize it. The λ value is set to 0.5. We used Adam optimizer [36] to train our generator and RMSProp optimizer [37] to train the discriminator. The K variable shown in Table I is 129 for 2x experiments and 65 for 4x experiments. The frequency offset value is calculated according to the following formula:

$$C = floor(\frac{K}{10}) + 1, \qquad (8)$$

where K is the number of frequency bins in the input spectrogram. The N variable shown in Table I is 141 and 199 for



(a) 2x SSR results

(b) 4x SSR results

Fig. 4: Spectrogram examples for 2x and 4x, shown in (a) and (b), respectively. The samples are randomly selected from the WSJ0 corpus (unseen speakers). The first row in each Figure shows the ground truth high-frequency range spectrograms. The second and third rows show the generated high-frequency range spectrograms of the proposed network trained with only the LSD loss (second rows) and with both LSD and GAN losses (third rows).

TABLE II: The objective evaluation results for 2x and 4x SSR experiments. The bolded values show the best results. Our method (*SSR-GAN*) outperforms the baselines for all metrics. *LSD HF* shows the LSD value calculated only for the high-frequency range, where *LSD Full* shows the LSD value calculated for the whole spectrogram.

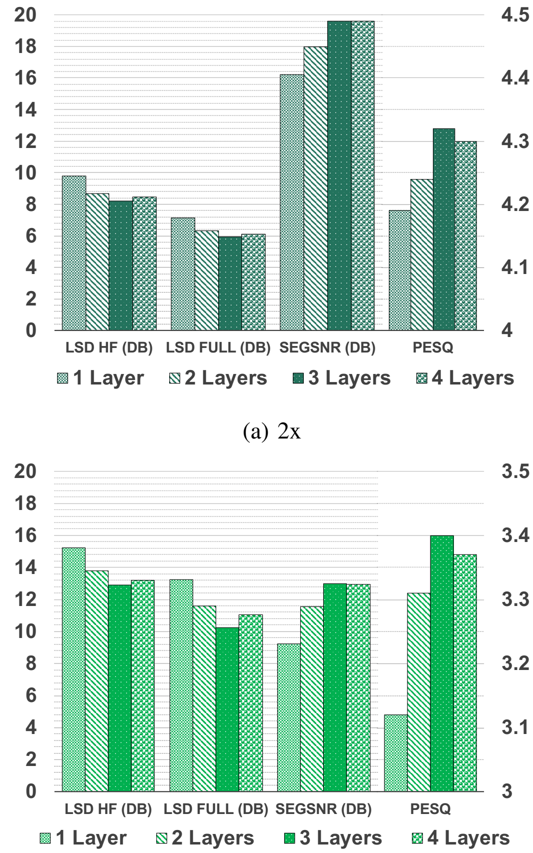
Scale	Method	LSD HF (dB)	LSD Full (dB)	SegSNR (dB)	PESQ
2x	BL1 [6]	9.32	7.06	15.73	4.21
	BL2 [7]	10.56	7.64	14.96	4.19
	SSR-LSD	8.60	6.09	17.58	4.25
	SSR-GAN	**8.20**	**5.95**	**19.64**	**4.32**
4x	BL2 [7]	16.20	14.96	8.24	2.89
	SSR-LSD	14.10	12.42	11.78	3.26
	SSR-GAN	**12.90**	**10.24**	**13.01**	**3.40**

2x and 4x super-resolution scales, respectively. The γ variable shown in Equation (5), which weighs the regularization term for the discriminator, is set to 2. Please note that we did not use decaying on this parameter as in the original work [8]. We normalized the input and output LPSs to have zero mean and unit variance. We calculated these statistics from the training data and applied them during inference. We reverted the normalization when we calculate the LSD loss during training since calculating LSD on normalized data does not make sense perceptually.

F. Results

Objective evaluation results are shown in Table II. The table shows the high-frequency LSD values (*LSD HF*), full-range frequency LSD values (*LSD Full*), SegSNR values and PESQ values for the baseline methods, our neural network trained with only the LSD loss (denoted as *SSR-LSD*) and that with the full loss *SSR-GAN*. *SSR-GAN* method outperforms the baselines in both 2x and 4x SSR tasks with a good margin in terms of all of the three objective evaluation metrics. The improvement of our method, compared to BL2, is more pronounced in the 4x setting.

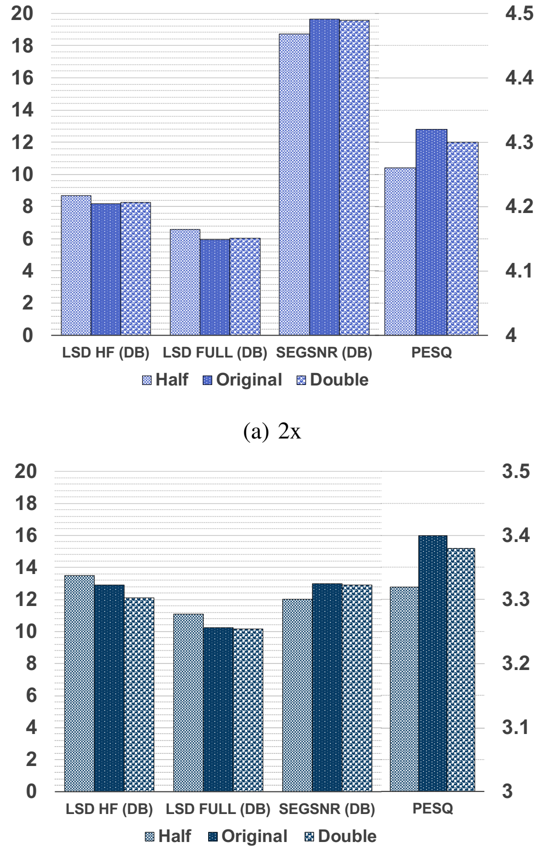
Figure 4 (a) and (b) show the example spectrograms, where the first row is the ground truth high-frequency range spectrogram, the second row is the high-frequency range spectrograms obtained from the *SSR-LSD*, and the third row shows *SSR-GAN* results, for 2x and 4x, respectively. Note that the LPSs on the second rows are overly smooth. After the GAN training, the resulting LPSs are sharper, containing fine details and usually, more energy. Generating more energy, in addition to generating fine details, leads to slightly better objective measures as seen in Table II. Nevertheless, the difference between the objective results for *SSR-LSD* and *SSR-GAN* are somewhat close compared to the baselines, especially for LSD metrics. We believe that the benefit of adversarial training is more evident for the subjective evaluations, which



(a) 2x

(b) 4x

Fig. 5: Objective evaluation results are presented for changing the number of layers in the encoder and decoder of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSNR*, and *PESQ* values, respectively.



(a) 2x

(b) 4x

Fig. 6: Objective evaluation results are presented for changing the number of filters of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. *Half* and *Double* means that the number of filters shown in Table I has been halved and doubled, respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSNR*, and *PESQ* values, respectively.

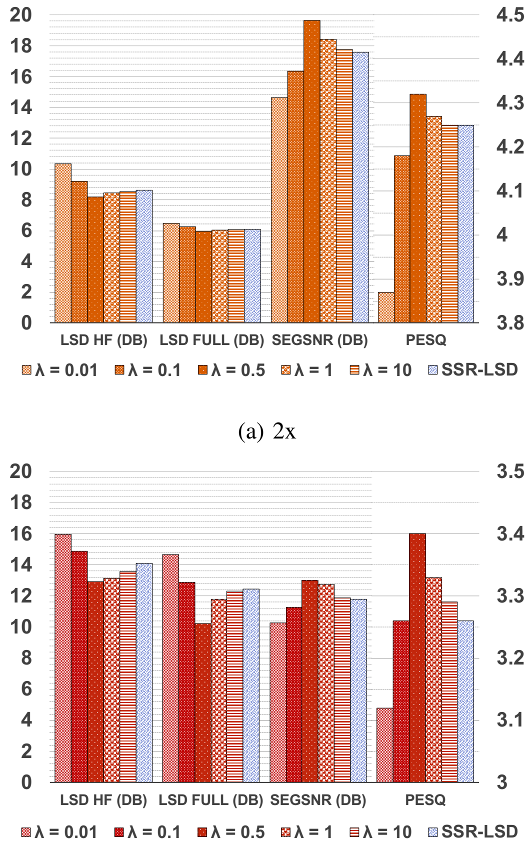
we discuss in Section IV-I.

G. Architecture and Parameter Analysis

In this section, we analyze our network by changing the number of hidden layers and the number of filters to see how they influence the objective evaluation results.

1) Number of Hidden Layers: Our proposed generator contains three encoder layers, followed by a bottleneck layer, three decoder layers, an upsampling layer, and an output layer. Note that the encoder and decoder layers are symmetric. We varied the number of layers in the encoder and decoder and reported the objective evaluation results in Figure 5. The results show that the network with three layers generally achieves the best performance across all of the objective metrics, although the differences between the three layers and four layers are rather small for the 2x scale. The network with one or two layers, however, achieves significantly worse performance. We believe that the networks with one or two layers perform worse due to underfitting, i.e., the capacity of these networks is not sufficient to learn patterns in the training corpus. As for the four-layer configuration, the performance slightly drops compared to three layers, which suggests that the increased capacity leads to overfitting. Considering the computational cost and slight performance differences between three layers and four layers, the three-layer configuration is preferred in our experiments.

2) Number of Filters: Next, we investigated the effect of varying the number of filters on our generator network. We investigated two other configurations in addition to the original configuration shown in Table I. The first configuration is called *Half*, where the number of filters of the original configuration is halved. The second one is called *Double* and has twice the number of filters of the original configuration. The results are shown in Figure 6. The results show that the configuration *Half* performs worse than the original in terms of objective measures, although the difference is not significant. This is a good option for systems with limited resources, where the number of filters can be halved in order to reduce the computational costs. Again, we suspect that the *Half* configuration suffers from underfitting due to the reduced capacity. For the *Double* configuration, increased computational complexity does not translate much into the



(a) 2x

(b) 4x

Fig. 7: Objective evaluation results are presented for different loss weight parameters (λ) and for *SSR-LSD* for comparison. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSNR*, and *PESQ* values, respectively.

performance gain compared to the original. Interestingly, for 4x scale, *Double* yields slightly better results for *LSD HF* and *LSD FULL* metrics, but overall, yields slightly lower speech quality. We believe that this is due to overfitting.

3) Loss Weight Parameter (λ): We analyzed the impact of changing the loss weight parameter λ. Increasing the value of λ increases the weight of the reconstruction loss. In this experiment, we used the following λ values: 0.01, 0.1, 0.5 (default), 1, and 10. The results for 2x and 4x scale experiments are shown in Figure 7. As the λ value increases the objective results get closer to the *SSR-LSD* results. On the other hand, decreasing λ from the default value of 0.5 leads to a degradation in generation quality. Since GAN loss becomes dominant, the generator produces speech-like spectrogram shapes that are unintelligible. In conclusion, we chose $\lambda = 0.5$ since it seemed a good balance between generating sharp and intelligible results.

H. Noise Analysis

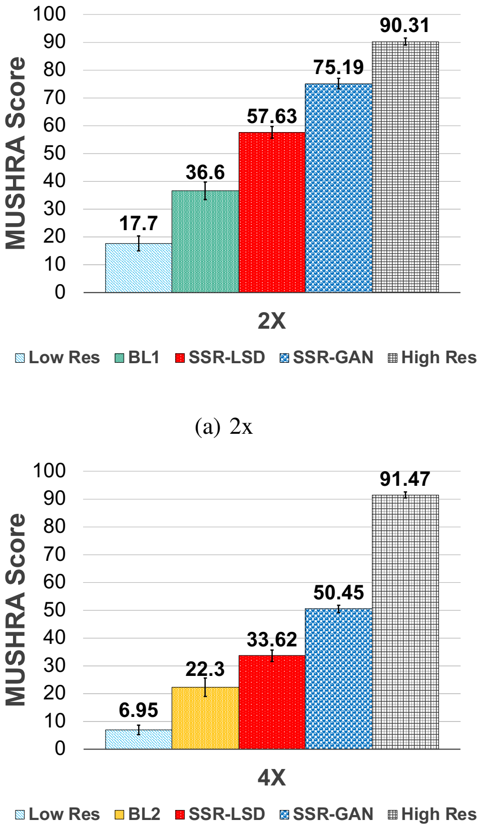
In real-world applications, the incoming speech signal has a high chance of containing background noise. Therefore, we further analyze our method against unseen time-varying noise types in this section. We trained our network against noise, by creating a dataset, where the narrowband signal is mixed with noise types in -6, -3, 0, 3, 6 and 9 dB SNR. We call this version of our network noise resilient *SSR-GAN* (*NR-SSR-GAN*). The network tries to predict the clean high-frequency range LPS from corrupted narrowband LPS. We employed the noise data from [38] for training. For evaluation, we used unseen noise types that were not present during training. Specifically, we used babble and factory noises described in [39] and a motorcycle noise described in [40]. We report the high-frequency range LSD results for samples that are mixed with 0 dB signal-to-noise ratio (SNR) testing noises using our base network model (*SSR-GAN*) and *NR-SSR-GAN* in Table III. The results suggest that noise resilient version of *SSR-GAN* can yield better scores against all three test noise types than the original *SSR-GAN*. The most challenging noise type is babble noise, followed by motorcycle noise and lastly, the factory noise.

TABLE III: Objective evaluation results for noise analysis.

Scale	Noise Type	Method	LSD HF (dB)
2x	Babble	SSR-GAN	14.63
		NR-SR-GAN	**10.23**
	Factory	SSR-GAN	13.47
		NR-SR-GAN	**9.97**
	Motorcycle	SSR-GAN	14.24
		NR-SR-GAN	**10.08**
4x	Babble	SSR-GAN	17.35
		NR-SR-GAN	**14.12**
	Factory	SSR-GAN	16.78
		NR-SR-GAN	**13.56**
	Motorcycle	SSR-GAN	17.16
		NR-SR-GAN	**13.84**

I. Subjective Evaluations

1) Perception Test: We conducted subjective evaluations to test if our method is successful regarding human perception. In our evaluations, we used a MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) test [41]. We compiled two test sets, one for 2x scale and one for 4x scale, where each of them contains 10 different tuples of signals with 5 signals in each tuple. These 5 signals included the narrowband signal (anchor), ground-truth high-resolution signal (reference), predicted super-resolution signals of our methods (*SSR-LSD* and *SSR-GAN*), *BL1* for 2x scale, and *BL2* for 4x scale. We wanted to limit the test time for each subject within 30 minutes; therefore we only used samples generated from one baseline method for each experiment. Before starting the experiments, each volunteer was trained by listening to 10 pairs of low and ground-truth high-resolution samples that were not contained in the testing tuples. After training, the testing utterances were presented to the volunteers in tuples, and within a tuple, the samples were presented randomly. The volunteers assigned a score between 0 and 100 for each utterance, where 0 corresponds to the low-resolution signal, and 100 corresponds to the high-resolution signal. We recruited 20 volunteers, where each of them evaluated 100 utterances (50 per 2x and 4x scales). During the test, the evaluators could listen



(a) 2x

(b) 4x

Fig. 8: The subjective evaluation results (MUSHRA test) for 2x and 4x scales are shown in (a) and (b), respectively. The error bars show the 95% confidence intervals.

to each utterance as many times as they wanted, and could listen to the reference signal (high-resolution signal) anytime. In MUSHRA experiments, the utterance tuples for which the evaluator failed to identify the hidden reference signal should be excluded. In our experiments, all evaluators successfully identified the hidden reference signal for all tuples.

The 2x scale experimental results are shown in Figure 8 (a). The ground-truth high-resolution speech has an average score of 90.31, which is followed by the *SSR-GAN* with an average score of 75.19%. The *SSR-LSD* achieves a score of 57.63%. The low-resolution signal and *BL1* has low scores, which are 17.7% and 36.6%, respectively. A paired t-test shows that the *SSR-GAN* score are statistically better compared to those of *SSR-LSD* and *BL1* at the significance level of 0.01 ($p =$ 10e-43).

Figure 8(b) shows MUSHRA test results for the 4x scale. The results show that the 4x experiments are more challenging compared to 2x experiments. The gap between the high-resolution score and the *SSR-GAN* is around 41%. *SSR-GAN* can still outperform the baseline method and has slightly more than 50% score. A paired t-test shows that the *SSR-GAN* results are statistically better compared to the *SSR-LSD* and *BL2* results at the significance level of 0.01 ($p =$ 10e-36).

Although *SSR-GAN* only slightly outperforms *SSR-LSD* in objective evaluation, their subjective evaluation results show a wider gap and the evaluators clearly preferred *SSR-GAN* over *SSR-LSD*. This outcome confirms the benefit of using the GAN loss for the SSR task.

TABLE IV: The intelligibility test results. The mean and standard deviation (std) of word error rate (WER) is shown for the 2x and 4x scale experiments using *SSR-GAN*.

Scale	Method	WER mean (%)	WER std (%)
2x	low-res	1.64	1.36
	SSR-GAN	1.48	1.28
4x	low-res	4.27	2.86
	SSR-GAN	3.82	2.12

2) Intelligibility Test: To rule out the possibility that the proposed *SSR-GAN* approach generates high-quality speech like sounds that are actually incomprehensible, we further conducted a listening test to check the intelligibility of the generated high-resolution speech. We employed the TIMIT dataset [42] for this test since it is distinct from our training dataset and the transcriptions of the sentences are available. As a baseline, we included the low-resolution samples into this test. We randomly selected 10 utterances with the low-resolution and selected 10 different utterances generated by *SSR-GAN* per 2x and 4x scales, totaling 40 sentences. We employed 20 volunteers among University of Rochester Graduate students, each of which evaluated all 40 sentences. During the experiments, the evaluators were presented each sample twice and were asked to transcribe the words.

Table IV shows the mean and standard deviation of the word error rate (WER) between the ground-truth and evaluators' transcription. The error rates for the 2x scale experiment are 1.48% and 1.64% for *SSR-GAN* and low-resolution signal (8 kHz sampling rate), and for the 4x scale experiment, they are 3.82% and 4.27% for *SSR-GAN* and low-resolution signal (4 kHz sampling rate). The 2x scale experiments have a lower error rate compared to 4x scale experiments since 8 kHz speech signals are more comprehensible than 4 kHz speech signals. Since *SSR-GAN* error rates are slightly lower than the low-resolution signal error rates, it can be concluded that the proposed SSR method does not impair the speech intelligibility.

J. Stability of GAN Training

In this study, we have considered different types of GANs and regularization techniques for stabilizing their training processing for SSR. We started from exploring the vanilla GAN [18]. After training it for a few epochs, it became unstable and produced nonsensical results. We observed similar issues for the WGAN [22] and the least-squares GAN [43]. Next, we explored GANs with regularization. WGAN-GP [23] and a GAN with instance noise regularization [24] produced more meaningful (spectrograms that looked like speech) yet not intelligible results. Finally, the regularization method suggested by Roth et al. [8] stabilized the GAN training, and led to the results obtained in this work. The regularizer [8] introduces a term that penalizes the weighted gradient-norm of the discriminator, leading to overcome the phenomenon

TABLE V: Computational complexity in terms of floating point operations per second (FLOPS), FLOPS per generating 1 second of speech and number of parameters for the baselines (BL1 and BL2) and the proposed SSR-GAN method.

Scale	**Method**	**Number of Parameters**	**Computational Complexity (FLOPS)**	**FLOPS per 1 second of speech**
2x	BL1 [6]	**11.2 M**	**45.1 M**	2.9 B
	BL2 [7]	56.4 M	76.2 B	202.7 B
	SSR-GAN	14.6 M	154.0 M	**616.0 M**
4x	BL2 [7]	56.4 M	76.2 B	202.7 B
	SSR-GAN	**16.0 M**	**190.5 M**	**762.0 M**

called mode collapsing effectively. Furthermore, it is a simple modification over the traditional GAN implementation and is computationally efficient compared to other regularization schemes.

K. Phase Estimation

In this work, we simply flipped the phase of the low-resolution signal as the phase of the high-frequency range of the SSR output. To improve our results, we considered Griffin-Lim algorithm [44] to estimate the phase of the high-range frequencies. However, the results contained artifacts, namely musical noise, and compared to flipped-phase we used in our experiments, they were not satisfactory. We think it is beneficial to share this finding with the research community. In addition, some example samples reconstructed with Griffin-Lim algorithm are shared in the link we provided. Future research directions to improve our results include estimating the phase using a deep learning approach or directly estimating the raw waveform.

L. Computational Complexity

We compare the computational performance of our method with the baselines using two metrics: floating point operations per second (FLOPS) and the number of trainable parameters. To obtain the FLOPS for each network, we employed Tensorflow's profiler.

Table V shows these values for 2x and 4x configurations of our method and the baselines. Please note that for *BL2*, the scale does not influence the computational complexity, since the input is always up-sampled to the target resolution. From values in the 2x scale, it can be observed that the fastest network during run time is *BL1*, followed by our method. It is important to highlight that *BL1* generates a single frame, while *BL2* and our method generate multiple frames. Therefore, we calculated the FLOPS value for generating 1 second of speech for each of these methods and concluded that our method has the lowest complexity.

V. Conclusion

We introduced a novel method for speech super-resolution using adversarial training and sequence-to-sequence modeling. To stabilize the GAN training, we employed a regularization method that penalizes the discriminator's gradient norms. Our generator architecture is a bottleneck encoder-decoder, while our discriminator architecture contains a convolutional decoder followed by fully connected layers. We used 1D kernels in the convolutional layers to reduce the computational complexity. The proposed method was evaluated for 2x (8 kHz to 16 kHz) and 4x (4 kHz to 16 kHz) scale super-resolution. We showed that our method outperforms the two state-of-the-art baseline methods in terms of objective metrics. We also conducted a subjective intelligibility evaluation, which showed that our method can score closely to the ground-truth high-resolution signal for the 2x scale, and can perform decently for the 4x scale. In additional experiments, we introduced a training method to increase the system's resilience against non-stationary, unseen noise types for real-world applications. Future directions include the estimation of phase information for better super-resolution quality.

References

[1] ITU, "Paired comparison test of wideband and narrowband telephony," in *Tech. Rep. COM 12-9-E*, Mar. 1993.

[2] L. J. Kepler, M. Terry, and R. H. Sweetman, "Telephone usage in the hearing-impaired population." *Ear and hearing*, vol. 13, no. 5, pp. 311–319, 1992.

[3] C. Liu, Q.-J. Fu, and S. S. Narayanan, "Effect of bandwidth extension to telephone speech recognition in cochlear implant users," *The Journal of the Acoustical Society of America*, vol. 125, no. 2, pp. EL77–EL83, 2009.

[4] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit." University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2016.

[5] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

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

[7] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[8] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Advances in Neural Information Processing Systems*, 2017, pp. 2018–2028.

[9] K.-Y. Park, "Narrowband to wideband conversion of speech using gmm based transformation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2000, pp. 1843–1846.

[10] B. Iser and G. Schmidt, "Bandwidth extension of telephony speech," in *Speech and Audio Processing in Adverse Environments*. Springer, 2008, pp. 135–184.

[11] H. Seo, H.-G. Kang, and F. Soong, "A maximum a posterior-based reconstruction approach to speech bandwidth expansion in noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6087–6091.

[12] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1. IEEE, 2001, pp. 665–668.

[13] P. Jax and P. Vary, "Artificial bandwidth extension of speech signals using mmse estimation based on a hidden markov model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1. IEEE, 2003, pp. I–I.

[14] G.-B. Song and P. Martynovich, "A study of hmm-based bandwidth extension of speech signals," *Signal Processing*, vol. 89, no. 10, pp. 2036–2044, 2009.

[15] 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.

[16] P. Smaragdis and B. Raj, "Example-driven bandwidth expansion," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*. IEEE, 2007, pp. 135–138.

[17] D. L. Sun and R. Mazumder, “Non-negative matrix completion for bandwidth extension: A convex optimization approach,” in *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*. IEEE, 2013, pp. 1–6.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[20] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[21] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for gans do actually converge?” in *International Conference on Machine Learning*, 2018, pp. 3478–3487.

[22] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International Conference on Machine Learning*, 2017, pp. 214–223.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[24] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, “Amortised map inference for image super-resolution,” *arXiv preprint arXiv:1610.04490*, 2016.

[25] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network.” in *CVPR*, vol. 2, no. 3, 2017, p. 4.

[26] A. Lucas, S. L. Tapia, R. Molina, and A. K. Katsaggelos, “Generative adversarial networks and perceptual losses for video super-resolution,” *arXiv preprint arXiv:1806.05764*, 2018.

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

[28] 3GPP2 C.S0014-C v1.0, “Enhanced variable rate codec, speech service option 3, 68 and 70 for wideband spread spectrum digital systems.”

[29] T. Gerkmann, M. Krawczyk, and R. Rehr, “Phase estimation in speech enhancementunimportant, important, or impossible?” in *IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*. IEEE, 2012, pp. 1–5.

[30] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[31] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[32] P. Mermelstein, “Evaluation of a segmental snr measure as an indicator of the quality of adpcm coded speech,” *The Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1664–1667, 1979.

[33] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,” *ITU-T Recommendation*, vol. 862, 2001.

[34] J. O. Smith. Digital audio resampling home page center for computer research in music and acoustics (ccrma). [Online]. Available: https://ccrma.stanford.edu/~jos/resample/

[35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[36] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[37] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[38] G. Hu, “100 nonspeech sounds,,” *Online: http://www. cse. ohio-state. edu/pnl/corpus/HuCorpus. html*, 2006.

[39] A. Varga and H. J. Steeneken, “Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems,” *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[40] Z. Duan, G. J. Mysore, and P. Smaragdis, “Speech enhancement by online non-negative spectrogram decomposition in nonstationary noise environments,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

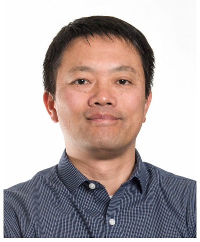
[41] I. Recommendation, “1534-1: Method for the subjective assessment of intermediate quality level of coding systems,” *International Telecommunication Union*, 2003.

[42] J. W. Lyons, “Darpa timit acoustic-phonetic continuous speech corpus,” *National Institute of Standards and Technology*, 1993.

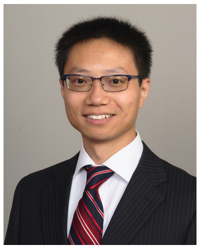
[43] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[44] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

S. Emre Eskimez attended Sabanci University and graduated with a Bachelor of Science degree in Mechatronics Engineering in 2011. He began graduate studies in the Department of Mechatronics Engineering at Sabanci University in 2011 and received a Master of Science degree in 2013. He began graduate studies in the Department of Electrical and Computer Engineering at the University of Rochester in 2014 and received a Master of Science degree in 2015. He pursued his research in speech processing and deep learning under the direction of Wendi Heinzelman and Zhiyao Duan. His research interests include speech processing, audio-visual signal processing, and deep learning.



Kazuhito Koishida received the B.E. degree in electrical and electronic engineering, and M.E., and Dr.Eng. degrees in intelligence science from Tokyo Institute of Technology, Tokyo, Japan, in 1994, 1995 and 1998, respectively. From 1998 to 2000 he was a Post-doctoral Researcher with Signal Compression Laboratory, University of California, Santa Barbara. He joined Microsoft Corporation, Redmond, USA, in 2000 and is currently a Principal Lead Scientist in Applied Sciences Group. His research interests include speech and audio processing, multimodal signal processing, and machine learning. He is a member of IEEE and ISCA.



Zhiyao Duan (S’09, M’13) is an assistant professor in the Electrical and Computer Engineering Department at the University of Rochester. He received his B.S. in Automation and M.S. in Control Science and Engineering from Tsinghua University, China, in 2004 and 2008, respectively, and received his Ph.D. in Computer Science from Northwestern University in 2013. His research interest is in the broad area of computer audition, i.e., designing computational systems that are capable of understanding sounds, including music, speech, and environmental sounds. He co-presented a tutorial on Automatic Music Transcription at ISMIR 2015. He received a best paper award at the 2017 Sound and Music Computing (SMC) conference and a best paper nomination at the 2017 International Society for Music Information Retrieval (ISMIR) conference.