applied sciences

MDPI

Article

Lightweight End-to-End Speech Enhancement Generative Adversarial Network Using Sinc Convolutions

Lujun Li *, Wudamu, Ludwig Kürzinger, Tobias Watzel and Gerhard Rigoll

Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany; wudamu@tum.de (W.); ludwig.kuerzinger@tum.de (L.K.); tobias.watzel@tum.de (T.W.); rigoll@tum.de (G.R.)
* Correspondence: lujun.li@tum.de

Abstract: Generative adversarial networks (GANs) have recently garnered significant attention for their use in speech enhancement tasks, in which they generally process and reconstruct speech waveforms directly. Existing GANs for speech enhancement rely solely on the convolution operation, which may not accurately characterize the local information of speech signals—particularly high-frequency components. Sinc convolution has been proposed in order to allow the GAN to learn more meaningful filters in the input layer, and has achieved remarkable success in several speech signal processing tasks. Nevertheless, Sinc convolution for speech enhancement is still an under-explored research direction. This paper proposes Sinc–SEGAN, a novel generative adversarial architecture for speech enhancement, which usefully merges two powerful paradigms: Sinc convolution and the speech enhancement GAN (SEGAN). There are two highlights of the proposed system. First, it works in an end-to-end manner, overcoming the distortion caused by imperfect phase estimation. Second, the system derives a customized filter bank, tuned for the desired application compactly and efficiently. We empirically study the influence of different configurations of Sinc convolution, including the placement of the Sinc convolution layer, length of input signals, number of Sinc filters, and kernel size of Sinc convolution. Moreover, we employ a set of data augmentation techniques in the time domain, which further improve the system performance and its generalization abilities. Compared to competitive baseline systems, Sinc–SEGAN overtakes all of them with drastically reduced system parameters, demonstrating its effectiveness for practical usage, e.g., hearing aid design and cochlear implants. Additionally, data augmentation methods further boost Sinc–SEGAN performance across classic objective evaluation criteria for speech enhancement.

Keywords: speech enhancement; generative adversarial networks; Sinc convolution; data augmentation; raw samples

Citation: Li, L.; Wudamu; Kürzinger, L.; Watzel, T.; Rigoll, G. Lightweight End-to-End Speech Enhancement Generative Adversarial Network Using Sinc Convolutions. *Appl. Sci.* **2021**, *11*, 7564. https://doi.org/10.3390/app11167564

Academic Editor: Emanuele Carpanzano

Received: 22 June 2021
Accepted: 14 August 2021
Published: 18 August 2021

1. Introduction

Speech enhancement is the task of removing or attenuating background noise from a speech signal, and it is generally concerned with improving the intelligibility and quality of degraded speech [1]. Speech enhancement is widely used as a preprocessor in speech-related applications including robust automatic speech recognition systems [2] and communication systems, e.g., speech coding [3], hearing aid design [4], and cochlear implants [5]. Conventional speech enhancement approaches include the Wiener filter [6], time–frequency masking [7–9], signal approximation [10,11], the spectral mapping [12,13], etc. In the last few years, supervised methods for speech enhancement which leverage deep learning methods have become mainstream. Well-known models include deep de-noising autoencoders [14], convolutional neural networks (CNNs) [15], and recurrent neural networks [16,17].

There exists a class of generative methods relying on generative adversarial networks (GANs) [18], which have CNNs as the backbone and have been verified to be efficient for speech enhancement [2,19–23]. GANs designate a generator for enhancement mapping

and a discriminator for distinguishing real signals from fake ones. With the transmitted information from the discriminator, the generator learns to produce outputs that resemble the realistic distribution of clean signals. Most past attempts deal with magnitude spectrum inputs, as it is often claimed that short-time phase is unimportant for speech enhancement [24]. However, further studies [25] demonstrate that a clean phase spectrum can deliver significant improvements to speech quality.

CNNs are the most popular deep learning architecture for processing raw speech inputs. This is due their weight sharing, local filters, and pooling features, which allow extraction of robust and invariant representations. The first convolutional layer is a critical part of waveform-based CNNs [26], since it processes high-dimensional inputs and extracts low-level speech representations for deeper layers. However, it is susceptible to vanishing gradient problems. To alleviate this issue, Ravanelli et al. [26] proposed the Sinc convolution to learn more meaningful filters in the input layer. Differently from a standard CNN, the Sinc-convolution layer convolves the waveform with a set of parametrized Sinc functions that implement band-pass filters, and only needs to learn the low and high cutoff frequencies. Consequently, the Sinc convolution is faster-converging and lightweight. It performs extraordinarily well in capturing selective speech clues, e.g., the pitch region, the first formant, and the second formant, which are essential for resembling clean speech signals. It has also achieved remarkable success in some fields of speech signal processing, e.g., speech recognition [27,28], speaker identification [26], keyword spotting [29], etc.

Unfortunately, Sinc convolution for speech enhancement is still an under-explored research direction. There is no available model fusing these two powerful paradigms—the Sinc convolution operating over raw speech waveforms and the generative adversarial models for speech enhancement. Therefore, we propose to bridge this gap by usefully merging the Sinc convolution and the speech enhancement GANs (SEGAN), resulting in a customizable, lightweight, and interpretable system, termed Sinc–SEGAN. Contributions of this paper are summarized as:

- Transfer the success achieved by the Sinc convolution in the field of speech and speaker recognition to the field of end-to-end speech enhancement.
- Optimize the SEGAN architecture from the seminal work [19], and enhance the original Sinc convolution layer to fit the advanced SEGAN.
- Analyze the learned filters of the Sinc convolution layer.
- Apply data augmentation techniques on raw speech waveforms directly to further improve the system performance.

Experimental results show that the proposed Sinc–SEGAN overtakes a set of competitive baseline models, especially on higher-level perceptual quality and speech intelligibility. Additionally, the system parameters reduce drastically; up to merely 17.7% of the baseline system. In addition, data augmentation techniques further boost the system performance across all classic objective evaluation metrics of speech enhancement. Analysis of the Sinc filters shows that the learned filters are tuned precisely to capture critical speech clues. Experimental results demonstrate the potential applications of the proposed system, e.g., portable devices, hearing aid design, and cochlear implants. Notably, the proposed Sinc–SEGAN system is generic enough to be applied to existing GAN models for speech enhancement to further improve performance.

2. Related Works

Pascual et al. [19] focus on generative architectures for speech enhancement, which leverage the ability of deep learning to learn complex functions from large example sets. The enhancement mapping is accomplished by the generator, whereas the discriminator, by discriminating between real and fake signals, transmits information to the generator so that the generator can learn to produce outputs that resemble the realistic distribution of clean signals. The proposed system learns from different speakers and noise types, and incorporates them together into the same shared parametrization, which makes the system simple and generalizable in those dimensions.

On the basis of [19], Phan et al. [30] indicate that all existing SEGAN systems execute the enhancement mapping via a single stage by a single generator, which may not be optimal. In this light, they hypothesize that it would be better to carry out multi-stage enhancement mapping rather than a single-stage one. To this end, they divide the enhancement process into multiple stages, with each stage containing an enhancement mapping. Each mapping is conducted by a generator, and each generator is tasked to further correct the output produced by its predecessor. All these generators are gradually cascaded to enhance a noisy input signal to yield an refined enhanced signal. They propose two improved SEGAN frameworks, namely iterated SEGAN (ISEGAN) and deep SEGAN (DSEGAN). In the ISEGAN system, parameters of its generator are fixed, constraining ISEGAN's generators to apply the same mapping iteratively, as its name implies. DSEGAN's generators have their own independent parameters, allowing them to learn different mappings flexibly. However, the parameters of DSEGAN's generators are N_{G} times more numerous than ISEGAN's generators, where N_{G} is the number of generators.

Phan et al. [31] reveal that the existing class of GANs for speech enhancement solely rely on the convolution operation, which may obscure temporal dependencies across the sequence input. To remedy this issue, they propose a self-attention layer adapted from non-local attention, coupled with the convolutional and deconvolutional layers of SEGAN, referred to as SASEGAN. Furthermore, they empirically study the effect of placing the self-attention layer at (de)convolutional layers with varying layer indices, including all layers (as long as memory allows).

As Pascual et al. [19] report, they open the exploration of generative architectures for speech enhancement to progressively incorporate further speech-centric design choices for performance improvement. This paper aims to further optimize SEGAN by fusing the powerful diagram: Sinc convolution. To our best knowledge, although Sinc convolution has achieved great success and been widely utilized in the some fields of speech signal processing, e.g., speech recognition [27] and speaker verification [26], its implementation on the speech enhancement task remains unexplored.

3. Speech Enhancement GANs

Given a dataset $\mathcal{X} = \{(\boldsymbol{x_1^*}, \tilde{\boldsymbol{x}}_\mathbf{1}), (\boldsymbol{x_2^*}, \tilde{\boldsymbol{x}}_\mathbf{2}), \cdots, (\boldsymbol{x_N^*}, \tilde{\boldsymbol{x}}_\mathbf{N})\}$ consisting of N pairs of raw signals—clean speech signal $\boldsymbol{x}^*$ and noisy speech signal $\tilde{\boldsymbol{x}}$, speech enhancement aims to find a mapping $f_\theta(\tilde{\boldsymbol{x}}) : \tilde{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{x}}$ to transform the raw noisy signal $\tilde{\boldsymbol{x}}$ to the enhanced signal $\hat{\boldsymbol{x}}$. θ contains the parameters of the enhancement network.

Conforming to GAN design [18], the generator learns an effective mapping that can imitate the real data distribution to generate novel samples related to those of the training set, i.e., $\hat{\boldsymbol{x}} = G(\tilde{\boldsymbol{x}})$. In contrast, the discriminator plays the role of a classifier which distinguishes real samples, coming from the dataset that the generator is imitating, from fake samples, made up by the generator. The discriminator guides θ towards the distribution of clean speech signals, by classifying $(\boldsymbol{x}^*, \tilde{\boldsymbol{x}})$ as real and $(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{x}})$ as fake. Eventually, the generator learns to produce enhanced signals $\hat{\boldsymbol{x}}$ good enough to fool the discriminator such that the discriminator classifies $(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{x}})$ as real. The objective function of SEGAN reads

$$\begin{aligned} \min_{\mathrm{Dis}} \mathcal{L}(\mathrm{Dis}) = & \frac{1}{2}\mathbb{E}_{\boldsymbol{x}^*, \tilde{\boldsymbol{x}} \sim p_{data}(\boldsymbol{x}^*, \tilde{\boldsymbol{x}})}[\mathrm{Dis}(\boldsymbol{x}^*, \tilde{\boldsymbol{x}}) - 1]^2 + \\ & \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z}), \tilde{\boldsymbol{x}} \sim p_{data}(\tilde{\boldsymbol{x}})}[\mathrm{Dis}(\mathrm{Gen}(\boldsymbol{z}, \tilde{\boldsymbol{x}}), \tilde{\boldsymbol{x}})]^2, \end{aligned} \tag{1}$$

$$\begin{aligned} \min_{\mathrm{Gen}} \mathcal{L}(\mathrm{Gen}) = & \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z}), \tilde{\boldsymbol{x}} \sim p_{data}(\tilde{\boldsymbol{x}})}[\mathrm{Dis}(\mathrm{Gen}(\boldsymbol{z}, \tilde{\boldsymbol{x}}), \tilde{\boldsymbol{x}}) - 1]^2 \\ & + \lambda \| \mathrm{Gen}(\boldsymbol{z}, \tilde{\boldsymbol{x}}) - \boldsymbol{x}^* \|_1, \end{aligned} \tag{2}$$

$\mathrm{Dis}(\cdot)$ is the discriminator module, $\mathrm{Gen}(\cdot)$ is the generator module, and $\boldsymbol{z}$ denotes a latent variable. Inspired by the effectiveness of the L_1 norm in the image manipulation domain [32,33], it is added to the generator loss function to encourage the generator to gain

more fine-grained and realistic results. The influence of the L_1 norm is regulated by the hyper-parameter λ. Pascual et al. [19] set $\lambda = 100$ empirically, and hence we take over this value in all experiments.

4. Sinc-Convolution

Different from a standard convolutional layer that performs a set of time-domain convolutions between the input waveform and some Finite Impulse Response filters, Sinc convolution conducts the convolutional operation with a predefined function g, depending on few learnable parameters ϕ as

$$y[n] = x[n] * g[n, \phi], \tag{3}$$

where $x[n]$ is a chunk of speech waveforms, and $y[n]$ is the filtered output. Inspired by standard filtering in digital signal processing, Ravanelli et al. [26] define g as a filter bank consisting of rectangular bandpass filters. In the frequency domain, the magnitude of a generic bandpass filter can be written as

$$G[f, f_1, f_2] = rect(\frac{f}{2f_2}) - rect(\frac{f}{2f_1}), \tag{4}$$

where f_1 and f_2 are the learnable low and high cutoff frequencies, and $rect(\cdot)$ is the rectangular function in the magnitude frequency domain. In the time domain, the reference function g transforms to

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n). \tag{5}$$

The Sinc function here is the unnormalized Sinc function, i.e., $sinc(x) = \frac{sin(x)}{x}$. Ravanelli et al. [26] initialize filters with cutoff frequencies of the mel-scale filterbank, which has the advantage of directly allocating more filters in the lower part of the spectrum. Fainberg et al. [34] execute experiments over different initialization schemes, but no benefit for the downstream task is observed. Please note that there are two constraints in Equation (5) that need to be satisfied: $f_1 \geqslant 0$ and $f_2 \geqslant f_1$. In addition, the de facto frequences that are calculated in Equation (5) are f_1' and f_2', where

$$\begin{aligned} f_1' &= |f_1|, \\ f_2' &= |f_1| + |f_2 - f_1|. \end{aligned} \tag{6}$$

To smooth out the abrupt discontinuities at the end of g, Hamming window $w[n]$ [35] of length L is deployed by

$$g_w[n, f_1', f_2'] = g[n, f_1', f_2'] \cdot w[n], \tag{7}$$

where

$$w[n] = 0.54 - 0.46 \cdot cos(\frac{2\pi n}{L}). \tag{8}$$

It is also suggested that no significant performance difference appears when adopting other window functions [26]. As the filters g are symmetric, the filters can be computed efficiently by considering one side of the filter and inheriting the results for the other half. Moreover, the symmetry does not introduce any phase distortion, keeping the essence of processing raw inputs for speech enhancement.

Another remarkable property of Sinc convolution is its small parameter scale. If a CNN layer is composed of F filters of length M, a standard CNN employs $F \times M$ parameters. Unlike a standard convolutional layer, only two parameters are employed for each Sinc filter, regardless of its length. For instance, if $F = 64$ and $M = 251$, the CNN layer employs approximately 16 K parameters. By contrast, a Sinc-convolution layer only

employs 128 parameters. This property offers the possibility to obtain selective filters with many taps, with negligible parameter increment.

5. Sinc–SEGAN Architecture

We investigate two different deployments of the Sinc convolution illustrated in Section 4: (i) sitting before the first layer of the generator's encoder and the discriminator, and behind the last layer of the generator's decoder, referred to as the addition architecture, and (ii) acting as a substitute for the first standard convolutional layer of the generator's encoder and the discriminator, and the last standard convolutional layer of the generator's decoder, referred to as the substitution architecture.

In the case of the addition architecture, the generator makes use of an encoder–decoder architecture. The first layer of the encoder is Sinc convolution, using 64 filters of length 251 and stride = 1, followed by a max pooling layer (stride = 2). Thereafter, there are five one-dimensional strided convolutional layers with a common filter width of 31 and stride = 4, each followed by parametric rectified linear units (PReLUs) [36]. At the 6th layer of the encoder, the encoding vector $\boldsymbol{c} \in \mathbb{R}^{8\times1024}$ is stacked with the noise sample $\boldsymbol{z} \in \mathbb{R}^{8\times1024}$, sampled from the distribution $\mathcal{N}(0, I)$, and presented to the decoder.

The decoder component mirrors the encoder architecture, with the same number of filters and filter width, to reverse the encoding process through deconvolutions, namely fractional-strided transposed convolution. The last layer of the decoder is also a Sinc convolution (filter number = 64, kernel size = 251, and stride = 1), and there is an unmaxpooling layer (stride = 2) before it for upsampling. Learnable skip connections are deployed to connect the encoding layer with its corresponding decoding layer to allow the information to be summed to the decoder feature maps. The learnable vectors $\boldsymbol{a}_l$ multiply every channel of its corresponding shuttle layer l by a scalar factor $\gamma_{l,k}$. Hence, for the input of the jth decoder layer $\boldsymbol{h}_j$, we have the addition of the corresponding lth encoder layer response, following

$$\boldsymbol{h}_j = \boldsymbol{h}_{j-1} + \boldsymbol{a}_l \odot \boldsymbol{h}_l, \tag{9}$$

where $\odot$ is an element-wise product along channels.

The discriminator is constructed of a similar architecture to the encoder of the generator. However, it receives a two-channel input, i.e., $(\boldsymbol{x}^*, \tilde{\boldsymbol{x}})$ or $(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{x}})$, and utilizes virtual batch-norm [37] before LeakyReLU [38] activation with $\alpha = 0.3$. Moreover, the discriminator is topped up with a 1×1 convolutional layer to reduce the dimension of the output of the last convolutional layer for the subsequent classification task by the softmax layer. To sum up, in the addition architecture, the generator consists of 12 layers and the discriminator consists of six layers. We illustrate the addition architecture of Sinc–SEGAN in Figure 1.

The substitution architecture, as illustrated in Figure 2, is similar to the first case, but the original first layer of the encoder is substituted as Sinc convolution, using 64 filters of length 251. Its stride = 4, in line with the standard convolutional layer, and the max pooling layer is removed. Thereafter, there are four one-dimensional strided convolutional layers with a common filter width of 31 and stride = 4, each followed by PReLUs [36]. At the 6th layer of the encoder, the encoding vector $\boldsymbol{c} \in \mathbb{R}^{16\times1024}$ is stacked with the noise sample $\boldsymbol{z} \in \mathbb{R}^{16\times1024}$, sampled from the distribution $\mathcal{N}(0, I)$, and presented to the decoder.

The decoder component still uses the mirror architecture to reverse the encoding process through deconvolutions. The last deconvolutional layer of the decoder is also replaced with a Sinc convolution (filter number = 64, kernel size = 251, and stride = 4), and the upsampling layer is not needed anymore.

The first layer of the discriminator is also substituted with the Sinc convolution layer with 64 filters of length 251 and stride = 1. In summary, in the substitution architecture, the generator consists of 10 layers and the discriminator consists of five layers.

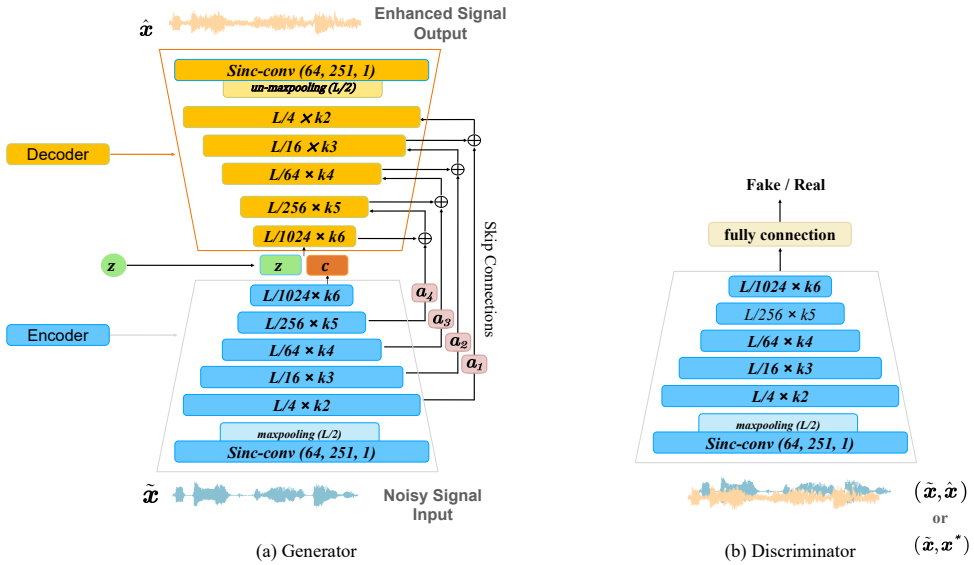

Figure 1. Illustration of the addition architecture of (**a**) the generator and (**b**) the discriminator, where the Sinc convolution sits before the first layer of the encoder and the discriminator, and behind the last layer of the decoder. Skip connections with learnable a_l are depicted in pink boxes, which are summed to each intermediate activation of the decoder.

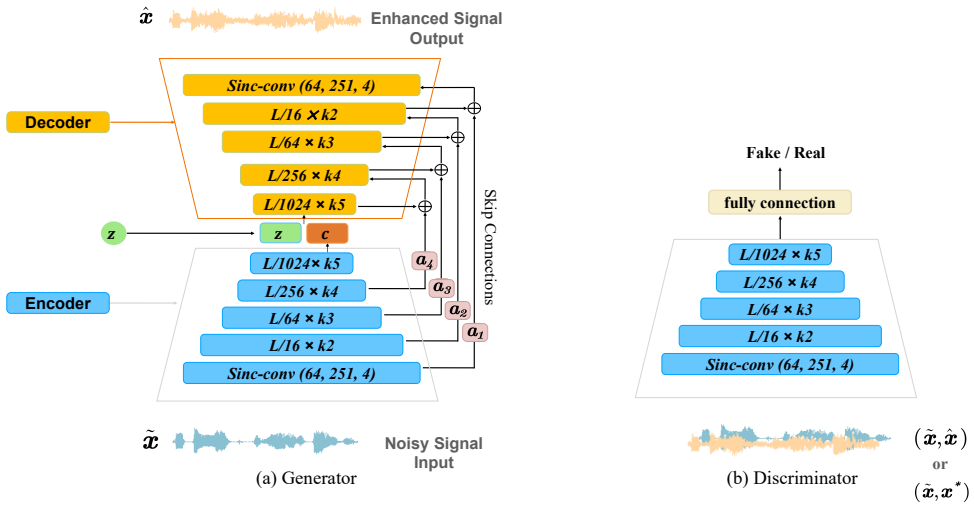

Figure 2. Illustration of the substitution architecture of (**a**) the generator and (**b**) the discriminator, where the Sinc convolution acts as the substitute of the first standard convolutional layers of the encoder and the discriminator, and the last standard convolutional layer of the decoder. Skip connections with learnable a_l are depicted in pink boxes, which are summed to each intermediate activation of the decoder.

6. Experimental Setup

6.1. Database

To assess the performance of proposed Sinc–SEGANs, we report objective measures on the Valentini [39] benchmarks. This publicly available dataset is originated from the Voice Bank corpus [40], which contains 30 speakers, of which 28 speakers are included in the training set while the remaining 2 are included in the test set. For the noisy training set, 40 different noisy conditions are considered by combining 10 sorts of intrusions (2 artificially generated and 8 derived from the Demand database [41]) with 4 different signal-to-noise ratios (SNRs) each (15, 10, 5, 0 dB). For the test set, 20 different noisy conditions are considered by combining 5 sorts of intrusions (all originated from the Demand database)

with 4 SNRs (17.5, 12.5, 7.5, and 2.5 dB) each. There are 10 different utterances in each adverse condition per training speaker; while, per test speaker, there are 20 utterances in each condition. Notably, the test set is entirely unseen by the training set; that is, no overlap exists in either speakers or adverse conditions. All utterances are downsampled to 16 kHz. Table 1 demonstrates the data structure of the employed corpus [39].

Table 1. Data structure of the corpus [39].

Subset	Speaker	Utterances	Intrusions	SNR (dB)
Training	28	11,200	10	15, 10, 5, 0
Test	2	800	5	17.5, 12.5, 7.5, 2.5

6.2. Evaluation Metrics

We evaluate the performance of the proposed system on six classic objective evaluation criteria for speech enhancement (the higher the better):

- PESQ: Perceptual evaluation of speech quality, using the wide-band version recommended in ITU-T P.862.2 [42] (in the range of $[-0.5, 4.5]$)
- STOI: Short-Time Objective Intelligibility [43] (in the range of $[0, 100]$)
- SSNR: Segmental SNR [44] (in the range of $[0, +\infty)$)
- CBAK: Mean Opinion Score (MOS) prediction of the intrusiveness of background noises [45] (in the range of $[1, 5]$)
- CSIG: MOS prediction of the signal distortion attending only to the speech signal [45] (in the range of $[1, 5]$)
- COVL: MOS prediction of the overall effect [45] (in the range of $[1, 5]$)

All metrics compare the enhanced signal with its corresponding clean reference over the test set as defined in Section 6.1 (All criteria are calculated based on the implementation demonstrated in [1], and all implementations are published on https://www.crcpress.com/downloads/K14513/K14513_CD_Files.zip (accessed on 15 February 2021)).

6.3. Implementation Details

The networks are trained for 100 epochs with the batchsize 100. Different from previous works [19,30,31], we utilize the Adam optimizer [46] ($\beta_1 = 0$ and $\beta_2 = 0.9$), with the two-timescale update rule (TTUR) [47]. According to the recent successful approach to training GANs quickly and stably [48], we set the learning rate of the discriminator to 0.0004 and that of the generator to 0.0001, i.e., the discriminator has a four-times-faster learning rate to virtually emulate numerous iterations in the discriminator prior to updating the generator. We extract raw speech chunks of length $L = 16,384$ (approximately 1 s) over 50% overlap as the input, to avoid any speech distortion caused by handcrafted features. A high-frequency pre-emphasis filter of coefficient 0.95 is applied to each waveform chunk before presenting it to the networks, as this is proved to help correct with some high-frequency artifacts in the de-noising setup. During testing, raw speech chunks are extracted from testing utterances without overlap, and outputs are correspondingly de-emphasized and concatenated as the enhanced waveforms. The number of filters varies along with the depth. In the situation of the addition architecture, they are $\{64, 64, 128, 256, 512, 1024\}$, resulting in the output size of the feature map $\{8092 \times 64, 2048 \times 64, 512 \times 128, 128 \times 256, 32 \times 512, 8 \times 1024\}$. In contrast, in the instance of the substitution architecture, they are $\{64, 128, 256, 512, 1024\}$, resulting in the output size of the feature map $\{4096 \times 64, 1024 \times 128, 256 \times 256, 64 \times 512, 16 \times 1024\}$. We also conduct ablation tests on the influence of the input length, the filter number and the kernel size. All experiments are implemented in PyTorch [49].

6.4. Data Augmentation

We employ three data augmentation methods on the dataset. First, we apply a random shift between 0 and 4 s. Second, we shuffle the noises with one batch to form new noisy mixtures, termed ReMix. Third, we deploy a band-stop filter with a stop-band between

f_1' and f_2' (termed Band Mask), sampled to remove 20% of the frequencies uniformly in the mel scale, which is equivalent to the SpecAugment [50] used for the automatic speech recognition task, in the time domain.

6.5. Baseline

For comparison, we take the seminal work [19], and other SEGAN variants [30,31] that we introduced in Section 2 as baseline systems. For [30], we choose the results of the ISEGAN with two shared generators and the DSEGAN with two independent generators as baseline results (the situation of $N_G = 2$). This is done for two reasons. On one hand, the number of generators leads to exponential parameter incrementation. On the other hand, Phan et al. [30] indicate only a marginal impact of ISEGAN's number of iterations, and for DSEGAN depth larger than $N_G = 2$ no significant performance improvements are seen. In [31], detailed results regarding the influence of the placement of the self-attention layer in the generator and the discriminator are presented. We choose the average result of coupling the self-attention layer with a single (de)convolutional layer (referred to as *SASEGAN-avg*), and the result of coupling self-attention layers with all (de)convolutional layers (referred to as *SASEGAN-all*) to ensure a fair comparison. It is worth noting that it is stated in [31] that, compared to *SASEGAN-avg*, the results of *SASEGAN-all* are slightly further boosted, but these gains are achieved at the cost of increased computation time and memory requirements.

7. Results

7.1. Ablation Tests on the Configuration of Sinc Concolution

We empirically study the impacts of the placement of the Sinc convolution layer, the input length, the number of Sinc filters, and the kernel size of the Sinc convolution. Table 2 demonstrates the configuration of five ablation tests, and Table 3 shows their results across the criteria introduced in Section 6.2. After making a comparison between these experiments, we can draw the following conclusions:

- Increasing the number of Sinc filters degrades the system performance, since more filters introduce more system complexity, and the training becomes more difficult, accordingly.
- Decreasing the kernel size of the Sinc convolution deteriorates the performance since smaller kernel size limits the ability to extract representative speech clues.
- Systems benefit from longer input length due to more context information being included.
- The addition architecture outperforms the substitution architecture as the former is deeper.

These experimental results explain why we choose the length of input signals as 1 s, the number of Sinc filters as 64, and the kernel size of Sinc convolution as 251.

Table 2. The demonstration of different configurations of five ablation tests. *substitution* is shortened to *sub*, and *addition* is shortened to *add*.

Experiment	A	B	C	D	E
Architecture	sub	sub	sub	sub	add
Input length	1 s	1 s	1 s	250 ms	1 s
Number of Sinc filters	64	80	64	64	64
Kernel size of Sinc convolution	251	251	101	251	251

Table 3. Ablation test results over different configurations of Sinc convolution: system architecture, input length, number of Sinc filters, and kernel size of Sinc convolution.

Experiment	PESQ	CSIG	CBAK	COVL	SSNR	STOI
A	2.37	3.55	3.13	2.97	8.68	93.40
B	2.32	3.49	2.84	2.91	5.51	92.99
C	2.40	3.46	3.07	2.89	8.66	93.39
D	2.36	3.57	3.07	2.94	8.70	93.37
E	2.39	3.69	3.23	3.00	8.71	93.53

7.2. Performance and Parameter Comparisons with Baseline Systems

Table 4 demonstrates the performance and parameter comparisons between the proposed Sinc–SEGANs (+augment) and the previous SEGAN variants [19,30,31] on the Valentini [39] benchmark. These results indicate that the substitution architecture outperforms baseline systems on PESQ, CBAK, and COVL. Considering the designs of these criteria, the results suggest that for speech signals enhanced by Sinc–SEGAN-sub (*substitution* is shortened to *sub*), the general perceptive quality is higher, and they are reasonably comprehensive for users. Comparable results are achieved on CSIG, SSNR and STOI. Please note that although Sinc–SEGAN-sub underperforms DSEGAN [30] on CSIG and SSNR or SASEFAN-all [31] on STOI, it outperforms SEGAN [19], ISEGAN [30], and SASEGAN-avg [31] across all criteria. Additionally, the number of Sinc–SEGAN-sub parameters is merely 31.0% as compared to SEGAN, ISEGAN, or SASEGAN-avg, 29.4% compared to SASEGAN-all, and 17.7% compared to DSEGAN. In contrast, Sinc–SEGAN-add (*addition* is shortened to *add*) outperforms all baseline systems, with the parameter scale that is 71% as compared to SEGAN, ISEGAN, or SASEGAN-avg, 67% compared to SASEGAN-all, and 41% compared to DSEGAN. Moreover, data augmentation methods deliver further improvements, leading to the best performance across all evaluation metrics, without additional parameters. These results validate the efficacy of Sinc convolution.

Table 4. Results on objective metrics of the proposed systems (Sinc–SEGANs) against previous SEGAN variants using the Valentini benchmark [39]. The unit of the number of parameters (Params) is million (M).

Architecture	Params (M)	Metric					
		PESQ	CSIG	CBAK	COVL	SSNR	STOI
Noisy	—	1.97	3.35	2.44	2.63	1.69	92.10
SEGAN [19]	294	2.16	3.48	2.94	2.79	7.66	93.12
ISEGAN [30]	294	2.24	3.23	2.93	2.68	8.19	93.29
DSEGAN [30]	513	2.35	3.56	3.10	2.94	8.70	93.25
SASEGAN-avg [31]	295	2.33	3.52	3.05	2.90	8.08	93.33
SASEGAN-all [31]	310	2.35	3.55	3.10	2.91	8.30	93.49
Sinc–SEGAN-sub	91	**2.37**	3.55	**3.13**	**2.97**	8.68	93.40
Sinc–SEGAN-add	210	**2.39**	**3.69**	**3.23**	**3.00**	**8.71**	**93.53**
Sinc–SEGAN-add +augment	210	**2.86**	**3.87**	**3.66**	**3.15**	**8.87**	**94.96**

7.3. Ablation Tests on Augmentation Methods

In order to better understand the influence of different data augmentation methods on the overall performance, we execute ablation tests. We report system performance on all evaluation criteria for each of the methods in Table 5. Results suggest that each of these data augmentation methods contributes to overall performance. The time shift augmentation produces the most significant performance increment, and the Band Mask algorithm is

the second. Surprisingly, ReMix augmentation only shows a limited contribution to the overall performance.

Table 5. Ablation study over different data augmentation methods: ReMix, Band Mask (BM), and the time shift.

Sinc–SEGAN-Add	PESQ	CSIG	CBAK	COVL	SSNR	STOI
+BM	2.44	3.55	3.37	3.05	8.79	93.75
+BM, +ReMix	2.45	3.57	3.40	3.07	8.81	93.80
+BM, +ReMix, +shift	2.86	3.87	3.66	3.15	8.87	94.96

7.4. Interpretation of Sinc Convolution

Inspecting learned filters is a valuable practice that provides insights into what the network is actually learning. To this end, we visualize the learned low and high cutoff frequencies of Sinc-convolution filters in Figure 3. The green area demonstrates the low and high cutoff frequencies for each filter. In addition, we also give the corresponding results of mel filters (purple area) for comparison. We plot four examples of the learned Sinc filters of the proposed system in Figure 4. As shown in Figure 4, the learned Sinc filters are rectangular band-pass filters in the frequency domain, echoing its definition in Section 4. Furthermore, we exhibit examples of spectrograms of the speech signal enhanced by SEGAN [19], DSEGAN [30], SASEGAN-all [31], and the proposed Sinc–SEGAN-add, respectively, in Figure 5. As observed in Figure 3, the Sinc convolution learns a filter bank containing more filters with high cut-frequencies. In addition, a tendency towards a higher amplitude is noticeable, indicating an inclination of the Sinc convolution to directly process raw speech waveforms. As we can see from Figures 4 and 5, Sinc convolution is specifically designed to implement rectangular band-pass filters. Considering the speech waveform distribution in the time domain, the specific design makes Sinc convolution suitable for extracting narrow-band speech clues, e.g., the pitch region, the first formant, and the second formant, in accordance with the results of the seminal work [26].

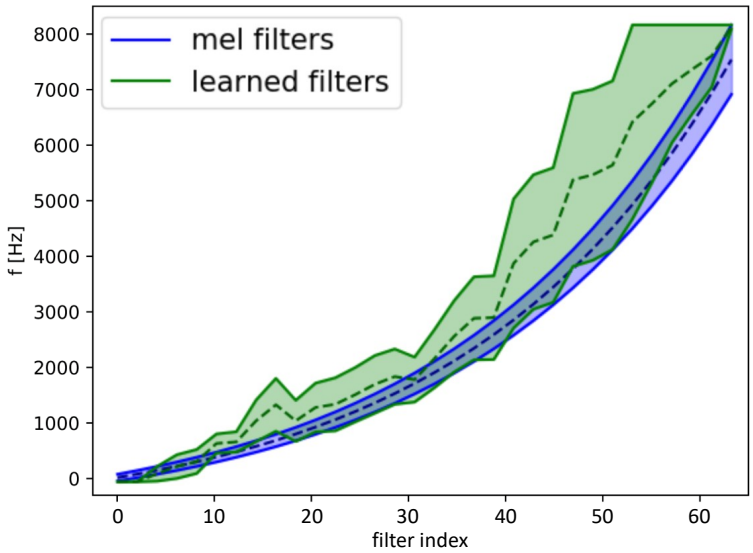

Figure 3. Visualization of the learned upper and lower bounds per Sinc-convolution filter.

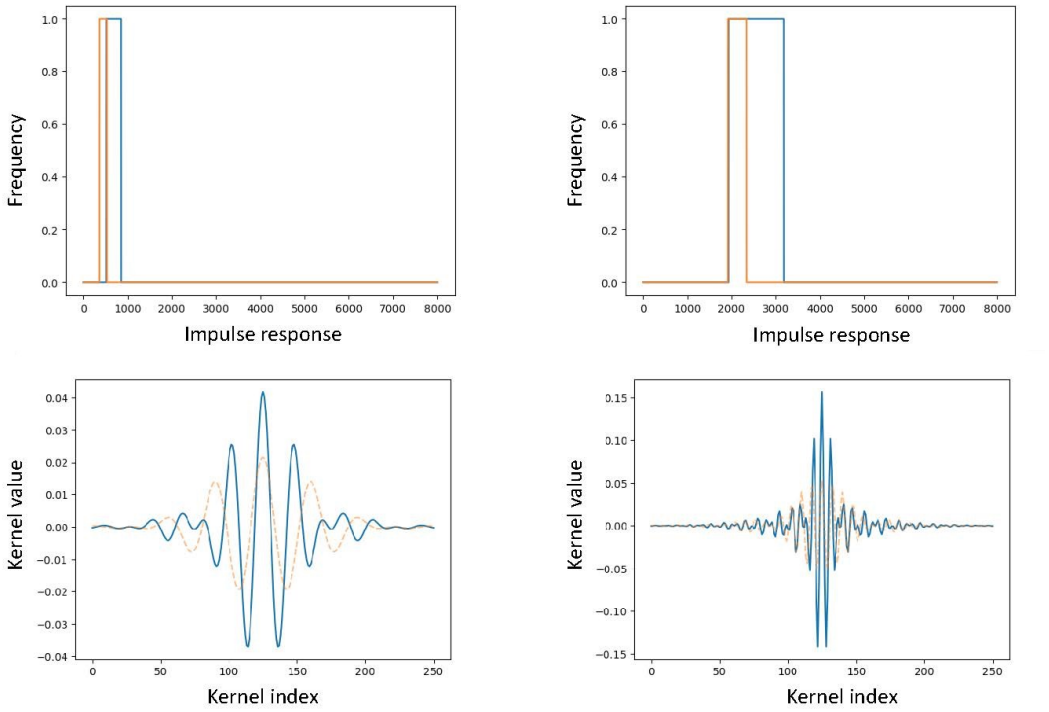

Figure 4. Examples of the learned filters of the Sinc convolution. The upper row reports the filters in the frequency domain, while the lower row reports their corresponding magnitude frequency response. The orange dashed line depicts the corresponding mel-scale filter.

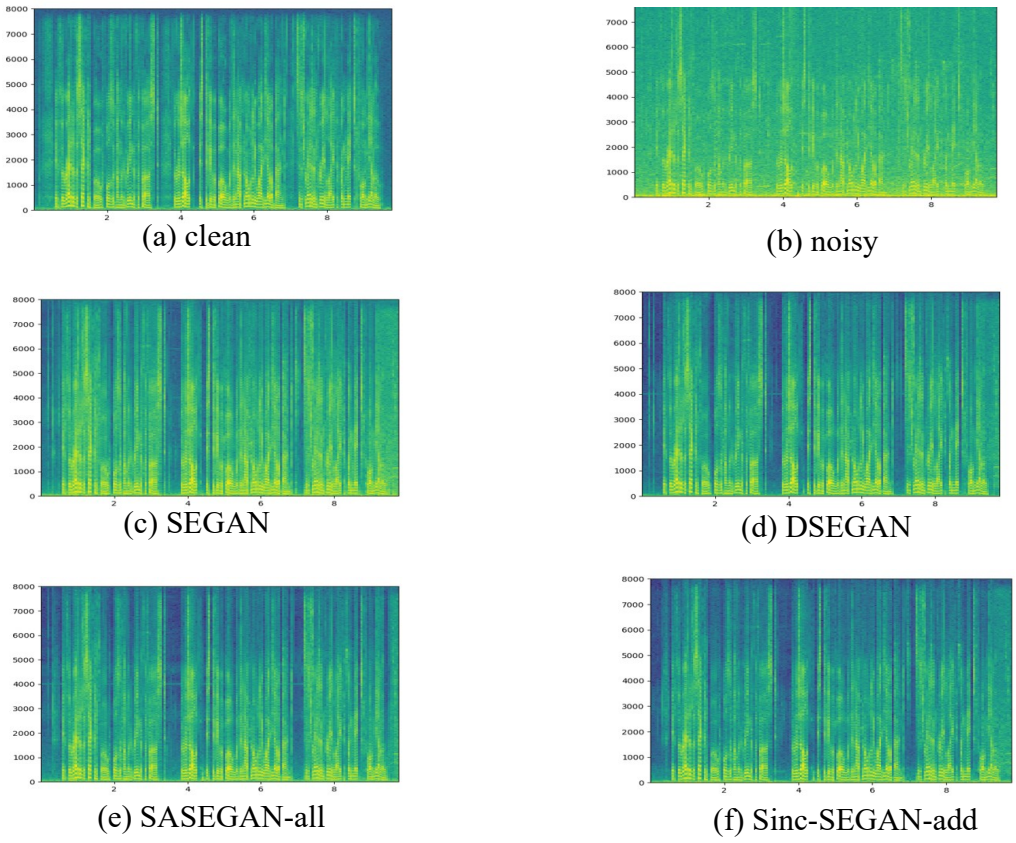

Figure 5. Spectrograms of an example utterance enhanced by (**c**) SEGAN [19], (**d**) DSEGAN [30], (**e**) SASEGAN-all [31], and the proposed (**f**) Sinc–SEGAN-add. We also exhibit the (**a**) clean and (**b**) noisy spectrograms for reference.

8. Conclusions

This paper proposes Sinc–SEGAN, a system that merges the Sinc convolution layer with the optimized SEGAN to capture more underlying representative speech characteristics. Sinc–SEGAN processes raw speech waveforms directly to prevent distortion caused by imperfect phase estimation. We investigate two different deployments of the Sinc convolution: (i) sitting before the first layer of the encoder and the discriminator, and behind the last layer of the decoder, referred to as Sinc–SEGAN-add, and (ii) acting as a substitute of the first standard convolutional layers of the encoder and the discriminator, and the last standard convolutional layer of the decoder, referred to as Sinc–SEGAN-sub. Ablation tests are conducted for the influence of the input length, number of Sinc filters, and kernel size of Sinc convolution. To train the proposed system more efficiently, we also employ three data augmentation methods in the time domain. Experimental results show that Sinc–SEGAN-sub yields enhanced signals with higher-level perceptual quality and speech intelligibility, even with drastically reduced system parameters. By contrast, the proposed Sinc–SEGAN-add overtakes all baseline systems across all classic objective evaluation criteria, with up to ∼50% fewer parameters compared to the baseline system. Moreover, data augmentation methods further boost the system performance. Analysis of the Sinc filters reveals that the learned filter bank is tuned precisely to select narrow-band speech clues, and is hence suitable for speech enhancement tasks in the time domain. Our future effort will be devoted to applying the Sinc convolution to other classic speech enhancement models, to further mitigate the lack of its application in the field of speech enhancement.

Author Contributions: Conceptualization, L.L.; methodology, L.L.; software, L.L. and W.; validation, L.L.; formal analysis, L.L.; investigation, L.L. and W.; resources, L.L.; data curation, L.L.; writing—original draft preparation, L.L. and W.; writing—review and editing, L.L., W., L.K., T.W. and G.R.; visualization, L.L., W., L.K., T.W. and G.R.; supervision, G.R. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Technical University of Munich Publishing Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset [39] utilized by this paper is publicly available at the publisher webpage: https://datashare.ed.ac.uk/handle/10283/1942 (accessed on 5 February 2021).

Acknowledgments: The authors wish to thank the editors and reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
GAN	generative adversarial network
SEGAN	speech enhancement generative adversarial network
SNR	Signal-to-Noise Ratio
MOS	Mean Opinion Score
SSNR	segmental SNR
STOI	Short-Time Objective Intelligibility
CBAK	MOS prediction of the intrusiveness of background noises
CSIG	MOS prediction of the signal distortion attending only to the speech signal
COVL	MOS prediction of the overall effect
PESQ	perceptual evaluation of speech quality

References

1. Loizou, P. *Speech Enhancement: Theory and Practice*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2017.
2. Donahue, C.; Li, B.; Prabhavalkar, R. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5024–5028.
3. Zhao, Z.; Liu, H.; Fingscheidt, T. Convolutional neural networks to enhance coded speech. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *27*, 663–678. [CrossRef]
4. Reddy, C.K.A.; Shankar, N.; Bhat, G.S.; Charan, R.; Panahi, I. An individualized super-Gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device. *IEEE Signal Process. Lett.* **2017**, *24*, 1601–1605. [CrossRef]
5. Goehring, T.; Bolner, F.; Monaghan, J.J.; Van Dijk, B.; Zarowski, A.; Bleeck, S. Speech enhancement based on neural networks improves speech intelligibility in noise for cochlear implant users. *Hear. Res.* **2017**, *344*, 183–194. [CrossRef] [PubMed]
6. Lim, J.; Oppenheim, A. All-pole modeling of degraded speech. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 197–210. [CrossRef]
7. Narayanan, A.; Wang, D. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 7092–7096.
8. Wang, Y.; Narayanan, A.; Wang, D. On training targets for supervised speech separation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1849–1858. [CrossRef] [PubMed]
9. Nie, S.; Liang, S.; Xue, W.; Zhang, X.; Liu, W. Two-stage multi-target joint learning for monaural speech separation. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015; pp. 1503–1507.
10. Weninger, F.; Hershey, J.R.; Le Roux, J.; Schuller, B. Discriminatively trained recurrent neural networks for single-channel speech separation. In Proceedings of the 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Atlanta, GA, USA, 3–5 December 2014; pp. 577–581.
11. Erdogan, H.; Hershey, J.R.; Watanabe, S.; Le Roux, J. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 708–712.
12. Xu, Y.; Du, J.; Dai, L.R.; Lee, C.H. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *23*, 7–19. [CrossRef]
13. Nie, S.; Liang, S.; Liu, W.; Zhang, X.; Tao, J. Deep learning based speech separation via nmf-style reconstructions. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 2043–2055. [CrossRef]
14. Lu, X.; Tsao, Y.; Matsuda, S.; Hori, C. Ensemble modeling of denoising autoencoder for speech spectrum restoration. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
15. Fu, S.W.; Hu, T.Y.; Tsao, Y.; Lu, X. Complex spectrogram enhancement by convolutional neural network with multi-metrics learning. In Proceedings of the 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, Japan, 25–28 September 2017; pp. 1–6.
16. Weninger, F.; Eyben, F.; Schuller, B. Single-channel speech separation with memory-enhanced recurrent neural networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3709–3713.
17. Sun, L.; Du, J.; Dai, L.R.; Lee, C.H. Multiple-target deep learning for LSTM-RNN based speech enhancement. In Proceedings of the 2017 Hands-free Speech Communications and Microphone Arrays (HSCMA), San Francisco, CA, USA, 1–3 March 2017; pp. 136–140.
18. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *arXiv* **2014**, arXiv:1406.2661.
19. Pascual, S.; Bonafonte, A.; Serra, J. SEGAN: Speech enhancement generative adversarial network. *arXiv* **2017**, arXiv:1703.09452.
20. Higuchi, T.; Kinoshita, K.; Delcroix, M.; Nakatani, T. Adversarial training for data-driven speech enhancement without parallel corpus. In Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 16–20 December 2017; pp. 40–47.
21. Qin, S.; Jiang, T. Improved Wasserstein conditional generative adversarial network speech enhancement. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 1–10. [CrossRef]
22. Li, Z.X.; Dai, L.R.; Song, Y.; McLoughlin, I. A conditional generative model for speech enhancement. *Circuits Syst. Signal Process.* **2018**, *37*, 5005–5022. [CrossRef]
23. Pascual, S.; Serrà, J.; Bonafonte, A. Towards generalized speech enhancement with generative adversarial networks. *arXiv* **2019**, arXiv:1904.03418.
24. Wang, D.; Lim, J. The unimportance of phase in speech enhancement. *IEEE Trans. Acoust. Speech Signal Process.* **1982**, *30*, 679–681. [CrossRef]
25. Paliwal, K.; Wójcicki, K.; Shannon, B. The importance of phase in speech enhancement. *Speech Commun.* **2011**, *53*, 465–494. [CrossRef]

26. Ravanelli, M.; Bengio, Y. Speaker recognition from raw waveform with SincNet. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 1021–1028.
27. Ravanelli, M.; Bengio, Y. Interpretable convolutional filters with sincnet. *arXiv* **2018**, arXiv:1811.09725.
28. Parcollet, T.; Morchid, M.; Linares, G. E2E-SINCNET: Toward fully end-to-end speech recognition. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7714–7718.
29. Mittermaier, S.; Kürzinger, L.; Waschneck, B.; Rigoll, G. Small-footprint keyword spotting on raw audio data with sinc-convolutions. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7454–7458.
30. Phan, H.; McLoughlin, I.V.; Pham, L.; Chén, O.Y.; Koch, P.; De Vos, M.; Mertins, A. Improving GANs for speech enhancement. *IEEE Signal Process. Lett.* **2020**, *27*, 1700–1704. [CrossRef]
31. Phan, H.; Nguyen, H.L.; Chén, O.Y.; Koch, P.; Duong, N.Q.; McLoughlin, I.; Mertins, A. Self-Attention Generative Adversarial Network for Speech Enhancement. *arXiv* **2020**, arXiv:2010.09132.
32. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
33. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
34. Fainberg, J.; Klejch, O.; Loweimi, E.; Bell, P.; Renals, S. Acoustic model adaptation from raw waveforms with SincNet. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 897–904.
35. Rabiner, L.; Schafer, R. *Theory and Applications of Digital Speech Processing*; Prentice Hall Press: Hoboken, NJ, USA, 2010.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
37. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training GANs. *arXiv* **2016**, arXiv:1606.03498.
38. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML* **2013**, *30*, 3.
39. Valentini-Botinhao, C. *Noisy Speech Database for Training Speech Enhancement Algorithms and TTS Models*; University of Edinburgh, School of Informatics, Centre for Speech Technology Research: Edinburgh, UK, 2017.
40. Veaux, C.; Yamagishi, J.; King, S. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In Proceedings of the 2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), Gurgaon, India, 25–27 November 2013; pp. 1–4.
41. Thiemann, J.; Ito, N.; Vincent, E. The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings. In *Proceedings of Meetings on Acoustics ICA2013*; Acoustical Society of America: Montreal, QC, Canada, 2013; Volume 19, p. 035081.
42. Rec, I. *P. 862.2: Wideband Extension to Recommendation P. 862 for the Assessment of Wideband Telephone Networks and Speech Codecs*; International Telecommunication Union, CH: Geneva, Switzerland, 2005.
43. Taal, C.H.; Hendriks, R.C.; Heusdens, R.; Jensen, J. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 2125–2136. [CrossRef]
44. Quackenbush, S.R. Objective Measures of Speech Quality. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 1995.
45. Hu, Y.; Loizou, P.C. Evaluation of objective quality measures for speech enhancement. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *16*, 229–238. [CrossRef]
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv* **2017**, arXiv:1706.08500.
48. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
49. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
50. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv* **2019**, arXiv:1904.08779.