INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Personalized Dereverberation of Speech

Ruilin Xu[1], *Gurunandan Krishnan*[2], *Changxi Zheng*[1], *Shree K. Nayar* [12]

[1] Columbia University, [2] Snap Inc.

rxu@cs.columbia.edu, guru@snap.com, cxz@cs.columbia.edu, nayar@cs.columbia.edu

Abstract

Classic non-blind speech dereverberation methods produce high-quality results *only* when the precise impulse response is known. Alternatively, learning-based blind methods cannot ensure adequate dereverberation in *all* environments. We propose an *environment-* and *speaker-specific* approach combining the advantages of both approaches. With a simple, one-time personalization step, our model generalizes a single measured impulse response to its spatial neighborhood. Specifically, the two-stage model performs feature-based Wiener deconvolution followed by a network-based refinement. Extensive experimental results indicate that our approach quantitatively and qualitatively outperforms the state-of-the-art methods. Additional user studies confirm that our method is overwhelmingly favored by listeners.
Index Terms: speech enhancement, speech dereverberation, neural networks

1. Introduction

Reverberation or echo in recorded audio depends on the acoustics of the environment and the locations of the audio source and microphone within that environment [1]. This interplay is described by the room impulse response (RIR) [2]. When a reverberant audio recorded in one environment is played in another, the listener in the second environment may perceive the original audio as "out of place". However, if the remote audio were lacking reverberations or were dereverberated, then the listener would perceive the audio source as *in situ*. Hence, dereverberation can make applications like video conferencing acoustically intimate.

Dereverberation is a long-studied problem. Given an audio source signal x, the reverberant signal y received by the microphone can be modeled as,

$$y = x * h + n, \tag{1}$$

where h is the RIR between the source and the microphone, n is additive noise. Dereverberation aims to recover the clean signal x from the contaminated signal y.

When the precise RIR is known, classical methods like Wiener deconvolution [3] can remove the reverberation in the audio. However, during an online meeting, the user may move around leading to varying RIR. Even these small variations in RIR can compromise the performance of classical methods. Furthermore, it is impractical to measure RIR for all possible user locations. Hence, blind dereverberation methods [4–8] have been popular. Among them, the weighted prediction error (WPE) algorithm [9] is a well-established approach that iteratively estimates a filter to predict the current reverberation tail at each time frame. It is well-known that WPE can suppress late reverberations to a large extent. As a result, many extensions to WPE have been proposed [10–12], even those that combine WPE with learning-based methods [13, 14].

With the advancement of machine learning, network-based dereverberation methods [15–22] have achieved even better results. Unfortunately, despite the vast quantity of training data available today, these techniques are still limited in their ability to generalize to the enormous space of possible RIRs. We invite the reader to listen [1] to examples of existing dereverberation methods that reveal not only the persistence of some reverberation but also the muffling of higher frequencies.

In this paper, we focus on dereverberation from the perspective of online communication in common spaces such as offices, conference rooms, and lecture halls. We assume the users are in front of a computer (laptop, desktop, or video conference system) with a microphone and loudspeaker. Typically, two —the environment and the microphone—of the three factors that impact the RIR are relatively fixed, with the location of the user having the greatest impact on the RIR. We use this observation to design a simple, one-time personalization procedure. We measure the RIR for a single location of the user and then have them read a brief passage while moving within the space they are expected to occupy. Leveraging this data, we perform dereverberation by combing Wiener deconvolution and deep learning. This empowers our method to generalize the single RIR to the user's entire workspace. **We suggest the reader to evaluate the quality of our results and comparisons with previous work which are included here: https://dereverb.github.io.**

2. Method

We first describe the personalization step, in which we measure a single representative RIR and capture a few minutes of reverberant audio along with its corresponding clean audio. We then describe our dereverberation method that generalizes the representative RIR to the spatial neighborhood.

2.1. Representative RIR

Let the user stand roughly at the center of the space within which they would conduct online communication. The representative RIR (`rRIR`) is the RIR from the user at this location, denoted by P_{user}, to the location of the microphone (`env-mic`), denoted by $P_{\text{env-mic}}$. The typical method to measure RIRs requires the audio source to emit a sine-sweep and record it using the microphone. Please note that we choose the sine-sweep method (ESS) over the Maximum Length Sequence (MLS) because of its robustness as described in [23, 24]. However, emitting a sine-sweep is beyond human capability. Thus, we exploit the *reciprocity of RIR*.

Let $RIR(P_{\text{user}} \mid P_{\text{env-mic}})$ be the RIR from the user to `env-mic`. According to acoustic reciprocity, the RIR between a source and a receiver remains the same if the two are

[1] `https://dereverb.github.io`

10.21437/Interspeech.2023-2055

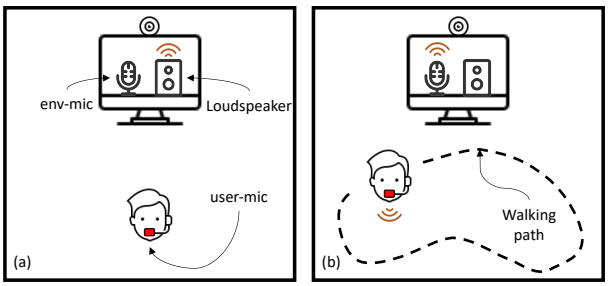

Figure 1: ***The personalization step.*** *(a) The user wears a head-worn microphone* `user-mic` *and stands approximately at the center of the usage region. The computer's loudspeaker emits a sine-sweep signal received by both* `user-mic` *and* `env-mic`*. The representative* ***RIR*** *(*`rRIR`*) is calculated from the signals captured by the two microphones. (b) The user then moves around the region while reading a brief passage, which is recorded by both microphones.*

swapped [25–27]. That is,

$$RIR(P_{\text{user}} \mid P_{\text{env-mic}}) = RIR(P_{\text{env-mic}} \mid P_{\text{user}}). \tag{2}$$

We require the user to wear a head-worn microphone (`user-mic`) during this phase as shown in Figure 1. Furthermore, we assume that the loudspeaker on the computer is roughly co-located with the `env-mic`. Then,

$$RIR(P_{\text{user}} \mid P_{\text{env-mic}}) \approx RIR(P_{\text{ls}} \mid P_{\text{user-mic}}), \tag{3}$$

where P_{ls} is the location of the loudspeaker and $P_{\text{user-mic}}$ is the location of the `user-mic`.

We now emit a sine-sweep over the loudspeaker and record it simultaneously by both the `env-mic` and the `user-mic`. `rRIR` is then calculated using the method described in [23]. In our implementation, the sine-sweep frequency ranges from 70 Hz to 20 kHz and is repeated three times for 2 seconds each.

While `rRIR` is accurate for the measured location, it can be an imprecise substitute for the spatial neighborhood.

2.2. Personalization

Since we cannot measure the RIR at every spatial location, we instead capture the *effects* of the spatially varying RIR. We ask the user to read a brief passage for five minutes while freely moving within the vicinity. We record the user's voice using both the `env-mic` and the `user-mic`. The audio recorded by `env-mic` has reverberations while the audio recorded by `user-mic` is treated as clean audio as its signal-to-reverberation ratio is very high.

2.3. Wiener Meets Deep Learning

Recall the reverberation model in Equation 1. If the precise RIR h and approximate noise characteristics are known, Wiener deconvolution [3] estimates the dereverberated signal $\hat{x}$ as,

$$\hat{x} = w * y, \tag{4}$$

where w is the Wiener filter. For practical reasons, this is usually computed in the frequency domain as,

$$\hat{X} = WY, \qquad W = \frac{H^*}{|H|^2 + \mathsf{NSR}}, \tag{5}$$

where $\hat{X}$, W, Y, and H are the Fourier transforms of $\hat{x}$, w, y, and h, respectively, and H^* is the complex conjugate of H. NSR is the expected noise-to-signal ratio, usually set empirically.

However, `rRIR` is only measured at one user location and cannot be generalized to the spatial neighborhood. Performing the naïve Wiener deconvolution of audio captured in the vicinity of `rRIR` leads to a substantial and irrecoverable loss in speech information. [2]

The data captured during personalization contains implicit information about how the RIR varies spatially. Leveraging this insight, our proposed dereverberation technique combines the advantages of both the Wiener filter and neural networks in a GAN framework [28]. Figure 2 illustrates our proposed model.

Our proposed generator $\boldsymbol{\mathcal{G}}$ accepts a reverberant signal y and `rRIR` $\hat{h}$ as inputs, extracts features of y in a high-dimensional latent space (feature extraction, $\boldsymbol{\phi}$), performs feature-based Wiener deconvolution (deep Wiener, $\boldsymbol{\omega}$), and finally refines and projects features back to a 1-dimensional signal (refinement, $\boldsymbol{\psi}$).

Feature extraction, $\boldsymbol{\phi}$: $\boldsymbol{\phi}$ takes a reverberant signal y as input, learns to extract useful features, and projects them to an n-dimensional latent space ($n = 32$ in our implementation), formulated as,

$$\boldsymbol{\phi}(y) = \left\{\boldsymbol{\phi}(y)_i\right\}_{i=1}^{n} : \mathbb{R}^1 \to \mathbb{R}^n. \tag{6}$$

Intuitively, we seek to extract representations of the audio signal that explicitly support Wiener deconvolution using `rRIR` $\hat{h}$, but also generalize to the spatial neighborhood.

This module consists of one Conv1d layer (kernel size $k = 15$, stride $s = 1$ and padding $p = 7$) with ReLU followed by two residual blocks ($k = 15$, $s = 1$ and $p = 7$).

Deep Wiener, $\boldsymbol{\omega}$: Partially inspired by [29], we propose our feature-based deep Wiener operation, which is analogous to Equation 4, as,

$$\boldsymbol{\omega}(y, \hat{h}) = \hat{w} * \boldsymbol{\phi}(y)_i, \quad \forall i = 1, \ldots, n. \tag{7}$$

Here we perform *channel-wise* Wiener deconvolution with $\boldsymbol{\phi}(y)_i$, each of the n extracted features of y, and $\hat{w}$, the Wiener filter obtained by utilizing $\hat{h}$, the imprecise `rRIR`. In frequency domain, $\hat{w}$ can be written as,

$$\hat{W} = \frac{\hat{H}^*}{|\hat{H}|^2 + \mathsf{NSR}}, \tag{8}$$

where $\hat{H}$ is the Fourier transform of $\hat{h}$. We use $\mathsf{NSR} = 0.1$ regardless of the noise level of the reverberant signal.

In short, this module deconvolves the input reverberant signal with `rRIR`, aiming to remove most of the reverberation existing in the signal. Since (1) `rRIR` is *not* the exact RIR used to generate the input signal, and (2) NSR is set at a *fixed* value, this process leads to residual reverberations and artifacts, which are handled by the refinement module.

Refinement, $\boldsymbol{\psi}$: Now that we obtain $\boldsymbol{\omega}(y, \hat{h})$, we perform a refinement process that enhances the signal in the latent space and then projects back to a 1-dimensional audio signal as,

$$\boldsymbol{\psi}\big(\boldsymbol{\omega}(y, \hat{h})\big) = \hat{x}. \tag{9}$$

Specifically, $\boldsymbol{\psi}$ is a U-Net-based multi-layer convolutional encoder and decoder with skip connections. The encoder gets as input the 32-dimensional signal from the previous module

[2] Please listen to the dereverberation results using naïve Wiener.

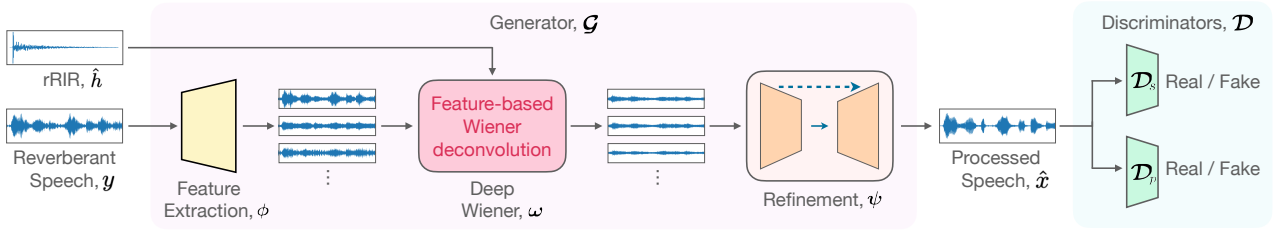

Figure 2: ***Proposed dereverberation network.*** *The generator $\mathcal{G}(y, \hat{h})$ accepts both a reverberant signal and an approximate RIR to perform feature-based Wiener deconvolution, which is then enhanced by the refinement module for removing any remaining noises and artifacts. The discriminator $\mathcal{D}$ contains a multi-scale discriminator $\mathcal{D}_s$, and a multi-period discriminator $\mathcal{D}_p$.*

and outputs its latent representation. Each of the 5 layers of the encoder consists of a Conv1d layer ($k = 8$, $s = 4$) followed by ReLU, another Conv1d layer ($k = 1$, $s = 1$), and finally a GLU activation. Then the latent representation is fed into a unidirectional LSTM network with 2 layers followed by a linear layer. Finally, the 5-layer decoder network outputs the dereverberated signal. Each decoder layer consists of a Conv1d layer ($k = 1$, $s = 1$) followed by a GLU activation, another Conv1d layer ($k = 8$, $s = 4$), and finally a ReLU function.

In summary, the proposed generator $\mathcal{G}$ described above and shown in Figure 2 can be mathematically described as,

$$\begin{aligned} \hat{x} &= \mathcal{G}(y, \hat{h}) \\ &= \psi\big(\boldsymbol{\omega}(y, \hat{h})\big) \\ &= \psi\Big[\Big(\hat{w} * \boldsymbol{\phi}(y)_i\Big)_{i=1}^{n}\Big]. \end{aligned} \tag{10}$$

Discriminators, $\mathcal{D}$: The discriminator module essentially evaluates whether the generated audio has reverberation or not. We use the multi-scale discriminator proposed in MelGAN [30], which evaluates audio samples at different scales. In our work, the discriminator operates at three scales: audio down-sampled by a factor of 1, 2, and 4. We also leverage the multi-period discriminator proposed in HiFi-GAN [31], which evaluates audio based on implicit structures at different periodic segments. As suggested by the paper, we set the periods to $[2, 3, 5, 7, 11]$.

Training Objectives: The objectives for our GAN model are:

$$\begin{aligned} \mathcal{L}_{\mathcal{G}} = \mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}) + \lambda_{FM}\mathcal{L}_{FM}(\mathcal{G}, \mathcal{D}) \\ + \lambda_{MEL}\mathcal{L}_{MEL}(\mathcal{D}), \end{aligned} \tag{11}$$

$$\mathcal{L}_{\mathcal{D}} = \mathcal{L}_{GAN}(\mathcal{D}, \mathcal{G}), \tag{12}$$

where $\mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D})$ and $\mathcal{L}_{GAN}(\mathcal{D}, \mathcal{G})$ are the standard GAN losses [28].

The feature matching loss, $\mathcal{L}_{FM}(\mathcal{G}, \mathcal{D})$, is a similarity metric between the clean and generated signals, defined as,

$$\mathcal{L}_{FM}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{T} \frac{1}{N_i} \|\mathcal{D}^{(i)}(\hat{x}) - \mathcal{D}^{(i)}(\mathcal{G}(y, \hat{h}))\|_1, \tag{13}$$

where $\mathcal{D}^{(i)}$ denotes the i-th layer feature map output of the discriminator; T is the number of layers in the discriminator; N_i denotes the number of features in the i-th layer.

The Mel-spectrogram loss, $\mathcal{L}_{MEL}(\mathcal{G})$, compares the spectrogram of the clean signal with that of the generated signal.

$$\mathcal{L}_{MEL}(\mathcal{G}) = \|\delta(\hat{x}) - \delta(\mathcal{G}(y, \hat{h}))\|_1, \tag{14}$$

where δ is the function that transforms a waveform to a Mel-spectrogram. This ensures that the generator produces realistic audio signals that match the clean ones across all frequencies.

3. Implementation

As described in Section 2.3, our model accepts as inputs a reverberant speech signal y, and an imprecise RIR captured in the vicinity of the speaker, $\hat{h}$. Both are in the waveform domain sampled at 16 kHz. The length of y is at least as long as h. To train the model in a supervised manner, we need a clean (noise and reverberation-free) copy of the reverberant signal. We take a two-stage approach to train the model: (1) pre-train with synthetic data and (2) fine-tune with personalization data.

Pre-training with Synthetic Data: We construct a large synthetic dataset using publicly available datasets with the following steps: (1) We randomly select 21,600 1-second speech clips (sampled at 16 kHz) for a total duration of 6 hours from LibriSpeech [32] as clean audio signals, of which 18,000 are sampled from LibriSpeech's training set and used to build our training set, and 3,600 are sampled from LibriSpeech's test set for testing. (2) RIRs (1-second long at 16 kHz) are sampled from the Aachen Impulse Response (AIR) dataset [33]. This dataset consists of real-world RIRs measured in various environments, such as a booth, office, meeting room, etc., with the reverberation time RT_{60} ranging from 0.08s to 0.83s. **We split the dataset by environments so that the set of environments for testing is different from the set for training.** For each environment, AIR provides multiple RIRs measured at different locations in close proximity, from which we randomly select two different RIRs – one to construct a synthetic reverberant signal and the other to use as the corresponding `rRIR`. Such RIR pairs are sampled for each of the 21,600 speech clips. (3) We also add background noises randomly chosen from the BUT Reverb dataset [34] (also at 16 kHz) with an SNR between 10 dB and 30 dB. As a result, we obtain 21,600 pairs of reverberant signals and `rRIRs` for training and testing without any overlap.

Pre-training using such a large synthetic dataset enables the model to generalize the input `rRIR` to its vicinity and treat dereverberation as a general problem.

Fine-tuning with Personalization Data: After pre-training with the synthetic data, we fine-tune (personalize) our network per user, per environment using user- and environment-specific reverberant signals and RIRs. We follow the steps below to get training data: (1) One `rRIR` is measured following the procedure in Section 2.1. This is used for both the training and testing tasks. (2) We collect a pair of the reverberant signal (recorded by the `env-mic`) and its corresponding clean signal (recorded by the `user-mic`) for each user within each environment for about 5 minutes (described in Section 2.2). (3) We collect an

Table 1: ***Ablation studies:*** *This table brings forth the impact of each network component. When Wiener deconvolution is disabled, the method reduces to blind dereverberation and quality suffers. When Wiener is enabled, but refinement is disabled, the quality jumps thereby proving the importance of Wiener deconvolution. We get the best quality when both are enabled. Evaluated on synthetic data.*

Wiener	Refinement	PESQ ↑	STOI ↑	SRMR ↑
×	✓	2.17	0.90	6.63
✓	×	2.10	0.89	7.38
✓	✓	**2.40**	**0.92**	**7.56**

Table 2: ***Robustness to location variation:*** *The dereverberation quality is optimal when the user is near the location of the measured* `rRIR`*. The quality degrades gracefully as the user moves further away. Evaluated on synthetic data.*

Distance	PESQ ↑	STOI ↑	SRMR ↑
0m	**2.21**	**0.92**	**8.78**
1m	2.12	0.90	8.37
2m	2.00	0.89	8.09
3m	1.74	0.83	7.06

additional 3 minutes of data for testing and reporting metrics. Please note that since the user is moving around during this step, the underlying RIR for the captured speech keeps varying and does not match the `rRIR` captured above. The user also reads different text excerpts during fine-tuning and testing and ensures that **the network never encounters testing data during the training phase** of the personalization step. Once the network has been fine-tuned with this data, the user no longer needs to wear the `user-mic`, and only the `env-mic` is used. With this **one-time** personalization process, our method is able to learn the implicit acoustic information of the user's voice and environment and produce high-quality dereverberation results when speaking.

In addition, to ensure data diversity for method comparison, we have recorded 4 individuals—two males and two females, native and non-native speakers—in 5 distinct environments, including a conference room, a tiled kitchen, a wood-floored bedroom, a glass-walled office, and a carpeted study room. Therefore, we have obtained $3 \times 4 \times 5 = 60$ minutes of real-world audio for method evaluation. Results are shown in Table 3.

4. Evaluation

We use the following widely used metrics to evaluate the quality of dereverberated audio. **i)** Perceptual Evaluation of Speech Quality (PESQ) [35], **ii)** Short-Time Objective Intelligibility (STOI) [36], and **iii)** Speech-to-Reverberation Modulation Energy Ratio (SRMR) [37].

4.1. Quantitative Evaluations

Table 3 provides the comparative evaluation of our method with classical/learning-based and blind/non-blind dereverberation methods. We report metrics for both synthetic and real data. To ensure a fair comparison, we fine-tune all learning-based methods with the recordings from the personalization step. For non-blind methods, the input `rRIR` from the personalization step is used. As highlighted, our method consistently outperforms other approaches for all the metrics. Specifically, based on SRMR scores, our method yields the least amount of remnant reverberation. Our method outperforms all other methods by an even greater margin on real-world data, demonstrating its robustness and practicality.

Table 3: ***Comparisons on synthetic/real dataset.*** *Our method yields the highest quality, most intelligible and least reverberant audio.*

Method	Blind	ML	PESQ ↑	STOI ↑	SRMR ↑
`Recorded`	-	-	1.8 / 1.6	0.9 / 0.7	6.9 / 4.3
`Wiener` [3]			1.3 / 1.6	0.8 / 0.7	4.7 / 4.6
`WPE` [4]	✓		1.9 / 1.7	0.9 / 0.7	7.4 / 5.9
`Demucs` [18]	✓	✓	1.5 / 1.0	0.9 / 0.6	7.3 / 4.7
`HiFi-GAN` [19]	✓	✓	2.1 / 1.7	**0.9** / 0.8	7.4 / 6.5
`Ours`		✓	**2.4 / 2.1**	**0.9 / 0.9**	**7.6 / 8.5**
`Clean`	-	-	4.6 / 4.6	1.0 / 1.0	8.0 / 8.6

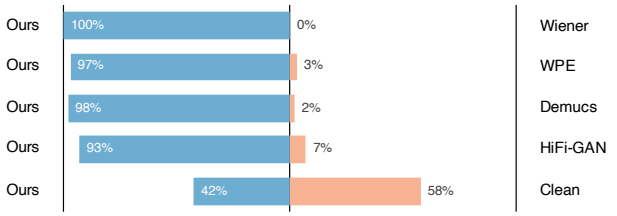

Figure 3: ***User study results.*** *We conduct a preference test between the dereverberation results of our method and those of the comparison techniques. Untrained listeners overwhelmingly favor our results. Even when compared to clean speech, our results are preferred nearly half of the time.*

4.2. Qualitative User Study

We conduct a user study to evaluate the perceptual quality of our method with untrained listeners. We randomly choose three clips from our dataset of real recordings and create pairs of results. Each pair contains audio processed by our method and by one of the comparison methods. Additionally, we create pairs of our result and the corresponding clean audio. In total, we create 15 pairs and invite 33 participants, including 18 male and 15 female, to choose a preferred audio within each pair in a blind study analogous to [19]. As shown in Figure 3 the listeners unambiguously favor the results from our method. It is also noteworthy that, compared to clean speech, our method is preferred nearly half of the time. This indicates that our method produces results that are generally perceived as clean as the original signals. **We invite the reader to listen to our results here: https://dereverb.github.io.**

5. Discussion

While our method produces state-of-the-art dereverberation results, we believe certain improvements can make it a practical go-to method. For online communications, real-time causal processing of audio is critical. Using a sliding window approach, our current implementation can run in real-time but is not computationally efficient on low-end laptops. Another drawback is that the method is *speaker-specific*. A *speaker-independent* fine-tuning will make our system more widely applicable. We intend to address these in the future.

6. References

[1] P. Howell, "Effect of speaking environment on speech production and perception," *Journal of the human-environment system : JHES*, vol. 11, pp. 51–57, 11 2008.

[2] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 2, pp. 145–152, 1988.

[3] N. Wiener, *Extrapolation Interpolation and Smoothing of Stationary Time Series*. The MIT Press, 1964.

[4] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, Oct 2018.

[5] B. Gillespie, H. Malvar, and D. Florencio, "Speech dereverberation via maximum-kurtosis subband adaptive filtering," vol. 6, 02 2001, pp. 3701–3704 vol.6.

[6] M. Tonelli, N. Mitianoudis, and M. Davies, "A maximum-likelihood approach to blind audio de-reverberation," pp. 2004–1, 11 2008.

[7] K. Lebart, J.-M. Boucher, and P. Denbigh, "A new method based on spectral subtraction for speech dereverberation," *Acta Acustica united with Acustica*, vol. 87, pp. 359–366, 05 2001.

[8] K. Furuya and Y. Kaneda, "Two-channel blind deconvolution for non-minimum phase impulse responses," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1997, pp. 1315–1318 vol.2.

[9] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time fourier transform representation," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 85–88.

[10] M. Parchami, W.-P. Zhu, and B. Champagne, "Speech dereverberation using linear prediction with estimation of early speech spectral variance," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 504–508.

[11] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[12] J. Caroselli, I. Shafran, A. Narayanan, and R. Rose, "Adaptive multichannel dereverberation for automatic speech recognition," 08 2017, pp. 3877–3881.

[13] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, "Frame-online dnn-wpe dereverberation," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018, pp. 466–470.

[14] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural Network-Based Spectrum Estimation for Online WPE Dereverberation," in *Proc. Interspeech 2017*, 2017, pp. 384–388.

[15] Y. Zhao, D. Wang, B. Xu, and T. Zhang, "Monaural speech dereverberation using temporal convolutional networks with self attention," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1598–1607, 2020.

[16] Z.-Q. Wang and D. Wang, "Deep learning based target cancellation for speech dereverberation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 941–950, 2020.

[17] Z. Wang, G. Wichern, and J. L. Roux, "Convolutive prediction for reverberant speech separation," *CoRR*, vol. abs/2108.07194, 2021.

[18] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," 2020.

[19] J. Su, Z. Jin, and A. Finkelstein, "Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks," 2020.

[20] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[21] J. Su, A. Finkelstein, and Z. Jin, "Perceptually-motivated environment-specific speech enhancement," in *ICASSP 2019*, May 2019.

[22] V. Kothapally, W. Xia, S. Ghorbani, J. H. Hansen, W. Xue, and J. Huang, "SkipConvNet: Skip convolutional neural network for speech dereverberation using optimally smoothed spectral mapping," in *Interspeech 2020*. ISCA, Oct 2020.

[23] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," *Journal of the Audio Engineering Society*, February 2000.

[24] P. Guidorzi, L. Barbaresi, D. D'Orazio, and M. Garai, "Impulse responses measured with mls or swept-sine signals applied to architectural acoustics: An in-depth analysis of the two methods and some case studies of measurements inside theaters," *Energy Procedia*, vol. 78, pp. 1611–1616, 2015, 6th International Building Physics Conference, IBPC 2015.

[25] P. Samarasinghe, T. D. Abhayapala, and W. Kellermann, "Acoustic reciprocity: An extension to spherical harmonics domain," *The Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. EL337–EL343, 2017.

[26] K. Wapenaar, "Unified matrix–vector wave equation, reciprocity and representations," *Geophysical Journal International*, vol. 216, no. 1, pp. 560–583, 10 2018.

[27] J. D. Achenbach, *Reciprocity in acoustics*, ser. Cambridge Monographs on Mechanics. Cambridge University Press, 2004, p. 55–69.

[28] 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.

[29] J. Dong, S. Roth, and B. Schiele, "Deep wiener deconvolution: Wiener meets deep learning for image deblurring," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1048–1059, 2020.

[30] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019.

[31] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, 2020.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5206–5210.

[33] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[34] I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[35] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq): A new method for speech quality assessment of telephone networks and codecs," vol. 2, 02 2001, pp. 749–752 vol.2.

[36] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.

[37] T. Falk, C. Zheng, and W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 1766–1774, 10 2010.