CAD-AEC: CONTEXT-AWARE DEEP ACOUSTIC ECHO CANCELLATION

Amin Fazel, Mostafa El-Khamy, Jungwon Lee

Samsung SoC Multimedia R&D, San Diego, USA

ABSTRACT

Deep-learning based acoustic echo cancellation (AEC) methods have been shown to outperform the classical techniques. The main drawback of the learning-based AEC is its dependency on the training set, which limits its practical deployment in mobile devices and unconstrained environments. This paper proposes a context-aware deep AEC (CAD-AEC) by introducing two main components. The first component of the CAD-AEC borrows ideas from the classical AEC and performs frequency domain adaptive filtering of the microphone signal, to provide the deep AEC network with features that have less dependency on the development context. The second component is a deep contextual-attention module (CAM), inserted between the recurrent encoder and decoder architectures. The deep CAM adaptively scales the encoder output during inference with calculated attention weights that depend on the context. Experiments in both matched and mismatched training and testing environments, show that the proposed CAD-AEC can robustly achieve better echo return loss enhancement (ERLE) and perceptual speech quality compared to the previous classical and deep learning techniques.

Index Terms— acoustic echo cancellation, deep learning, recurrent neural networks, gated recurrent unit, context awareness

1. INTRODUCTION

Acoustic echo is generated when a far-end user receives a modified version of his/her own voice due to acoustic coupling between a microphone and a loudspeaker at the near-end point. Acoustic echo cancellation (AEC) aims to eliminate the acoustic echo without distorting the desired near-end acoustic signal [1]. There has been renewed attention in AEC due to the explosive growth in mobile communications, smart speakers, hearables, etc.

Most classical AEC techniques estimate the acoustic path with an adaptive algorithm, and then subtract it from the microphone signal. The normalized least mean square (NLMS) algorithm [2] is highly popular for acoustic echo estimation due to its simplicity and robustness. Additionally, a residual echo suppressor (RES) is commonly applied to suppress the remaining residual echoes [3][4].

Supervised deep learning methods have recently been introduced to solve the AEC problem. Learning-based AEC have shown substantial improvements over the classical methods, when tested on environments similar to those they have been trained on. However, their performances drop dramatically when used in totally different contexts or conditions than the ones they were trained on. This limits their practical deployment in mobile devices which are not constrained to operate in a specific environment or context.

In this paper, we propose the context-aware deep AEC (CAD-AEC) to improve the robustness of learning-based AEC. CAD-AEC introduces two main components to make the deep AEC adaptive to changes in the deployment contexts and more robust to mismatches between the training and testing conditions. The first component which we call Adaptive Deep AEC borrows ideas from the classical AEC and devises adaptive frequency domain filtering of the near-end signal and the far-end signal to provide the deep AEC with adaptive features that are less dependent on the development context. This can be viewed as a hybrid approach combining the advantages of the classical AEC (dealing well with unseen environments) and the superior performance of the deep learning methods on learnt environments. The second component of CAD-AEC is the contextual attention module (CAM). In previous deep AEC approaches, the trained deep architectures apply the same kernels and weights to the input, regardless of the inference context. We introduce a recurrent encoder and decoder which take a large contextual window at the input. A contextual attention module is inserted between the encoder and the decoder. The CAM calculates the importance of the encoded features at each time step and scales them differently. Specifically, the introduced CAM adaptively maps the encoded contextual window with different context-aware weights to a hyperspace, and then estimated features of the near-end signal are decoded from the adapted encoded hyperspace.

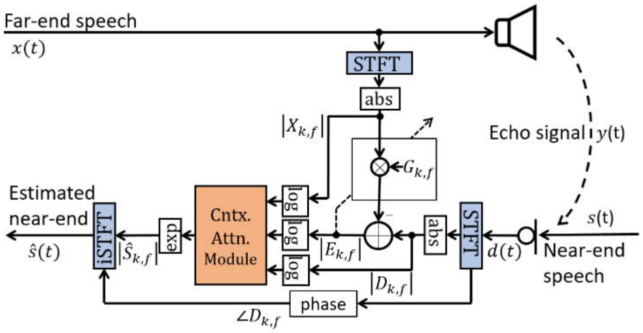


Fig. 1: Diagram of the proposed context-aware deep AEC.

Experiments conducted on both synthetic and real measured RIRs show that our proposed CAD-AEC method can cancel acoustic echo in both single-talk and double-talk periods, even with nonlinear distortions, while preserving the perceptual evaluation of speech quality (PESQ) score of the near end signal. The experiments also demonstrate the robustness of the proposed CAD-AEC in the case of mismatches between the training and testing data.

The remainder of this paper is organized as follows. Section 2 describes the related works. Section 3 provides details of our proposed method. Section 4 presents our experimental results. Finally, we summarize our findings in Section 5.

2. RELATED WORK

Neural networks have been used as a nonlinear RES in the past [5]. However, at the time, constraints in computational power and size of training data resulted in relatively small network implementation and limited overall performance. Recently, limited number of works have focused on improving the AEC performance by using deep learning models. Lee et al. [6] used deep feed-forward neural

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

networks as a nonlinear RES to suppress the remaining components after the AEC. Carbajal et al. [7] also proposed a neural network based RES with multiple inputs including the far-end speech, the AEC output, and the echo computed by the AEC. Zhang and Wang [8] proposed a deep learning based AEC method to predict a mask from features of the microphone and far-end signals, which is then used to resynthesize the near-end speech signal. Zhang et al. [9] used convolutional recurrent networks and long short-term memory (LSTM) to separate the near-end speech from the microphone recording. More recently we proposed deep recurrent neural networks with multitask learning to learn the auxiliary task of estimating the echo in order to improve the main task of estimating the near-end speech signal [10]. This work is different from the previously proposed methods as it combines the classical and deep learning techniques to create a contextual deep AEC to better adapt to the deployment context.

3. METHODOLOGY

3.1. Signal Model

Before describing our proposed method in detail, we begin by introducing our notation. We denote STFT complex-valued spectrum of an arbitrary time-domain signal $v(t)$ at frame k and frequency bin f as $V_{k,f}$. Its phase is denoted by $\angle V_{k,f}$ and its magnitude is denoted by $|V_{k,f}|$. Let $|V_k|$ be the vector of magnitudes at all frequency bins and frames k and $|\boldsymbol{V}| = [\,|V_{k-T}|, \ldots, |V_k|]$. The system model and schematic diagram of our method is illustrated in Fig 1. The microphone signal $d(t)$ consists of the near-end speech signal $s(t)$ and the acoustic echo $y(t)$:

$$d(t) = s(t) + y(t) \tag{1}$$

where acoustic echo is a modified version of the far-end signal $x(t)$ and includes room impulse response (RIR) and loudspeaker distortion. The goal of AEC is to generate the estimated near-end signal $\hat{s}(t)$ after removing any echo due to the far-end signal.

3.2. Adaptive Deep AEC

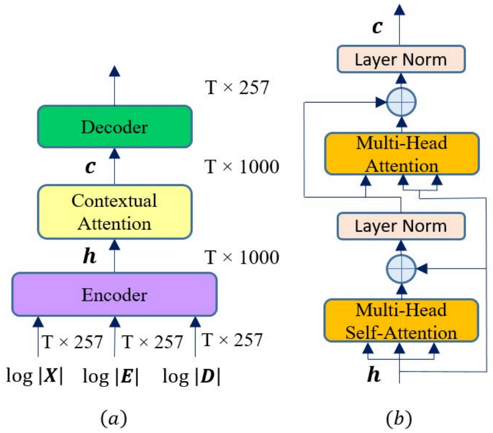
First, the short time Fourier transform (STFT) is applied on the microphone and far-end signals. Additionally, adaptive filtering is done in the frequency domain, where the normalized least mean square (NLMS) updating rule is used to estimate the acoustic path for each frequency bin. Then spectral error between the microphone and estimated acoustic echo signals is calculated. The proposed architecture estimates the near-end speech signal using the adaptive information extracted by the frequency domain NLMS (FDNLMS) updating rule.

We propose to estimate the near-end speech signal with deep encoder-decoder GRU networks. Specifically, we use the logarithmic spectral features of the far-end speech x, the microphone d, and the adaptive error signal e as inputs. The target output includes the logarithmic spectral features of the near-end speech signal s. In our proposed adaptive deep AEC method, we utilize the logarithmic spectral features of the error signal. We calculate the error signal for each frequency bin in the STFT domain as:

$$|E_{k,f}| = |D_{k,f}| - G_{k,f}|X_{k,f}|. \tag{2}$$

We use the adaptive NLMS algorithm to update the parameters of G as:

$$G_{k+1,f} = G_{k,f} + \frac{\mu}{P_{k,f}}|E_{k,f}||X_{k,f}| \tag{3}$$



(a) (b)

Fig. 2: Illustration of (a) the proposed contextual attention module and (b) attention mechanism.

where step size μ is normalized by the average power $P_{k,f}$ of the far-end signal and is obtained recursively by:

$$P_{k,f} = (1-\alpha)P_{k-1,f} + \alpha|X_{k,f}|^2 \tag{4}$$

where α is a forgetting factor between 0 and 1.

3.3. Contextual Attention Module

We use the causal context-aware inputs and outputs as described in [10]. During the training, each one of the contextual output frames are optimized against their own targets. In the inference time, the last frame is only considered as the output of the model. Fig. 2(a) illustrates the proposed contextual attention module. The encoder takes the concatenations of $\log|\boldsymbol{X}|$, $\log|\boldsymbol{E}|$, and $\log|\boldsymbol{D}|$ and map them to the hyperspace $\boldsymbol{h}$:

$$\boldsymbol{h} = \text{Encoder}(\log|\boldsymbol{X}|, \log|\boldsymbol{E}|, \log|\boldsymbol{D}|) \tag{5}$$

In our work, we use GRU for the encoder as it has a strong ability to model the sequences [11][12]. The encoder composed of one GRU layer with exponential linear unit (elu) activation. Then the contextual attention mechanism takes the encoder hyperspace and attends to the certain important region of the hyperspace:

$$\boldsymbol{c} = \text{Attention}(\boldsymbol{h}) \tag{6}$$

As shown in Fig. 2(b), the first layer in our attention mechanism is a multi-head self-attention (MHSA) layer [13]:

$$\boldsymbol{c}_{\text{MHSA}} = \text{softmax}(\frac{(\boldsymbol{h}\boldsymbol{W}_Q)(\boldsymbol{h}\boldsymbol{W}_K)^T}{\sqrt{d_h}})\boldsymbol{h}\boldsymbol{W}_V \tag{7}$$

where d_h is the dimension of hidden state $\boldsymbol{h}$ and $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ are learnable weights. A residual connection [14] is used around the MHSA layer followed by a layer normalization [15]. Additionally, a multi-head attention (MHA) layer is used where the queries for this layer is the output of the first layer normalization and the keys and values are the outputs of the encoder. Again a residual connection and a layer normalization are used around and after the MHA layer respectively. Finally, a stacked GRU with two layers takes the output of the attention layer to generate an estimate of the near-end signal in the logarithmic spectral space:

$$\log|\widehat{\boldsymbol{S}}| = \text{Decoder}(\boldsymbol{c}) \tag{8}$$

where *"elu"* and *"linear"* activations are used for the first and second layers of the decoder's GRU stacks respectively. The time domain signal is generated from the output of the decoder and the phase of the microphone signal using the inverse short time Fourier transform (iSTFT). For the loss function, we calculate the mean absolute error (MAE) between the ground-truth near-end speech s and the estimated output $\hat{s}$ in the logarithmic STFT feature domain over T=7 frames.

4. EXPERIMENTAL EVALUATION

4.1. Evaluation Metrics

The echo return loss enhancement metric (ERLE) is used to evaluate the echo reduction that is achieved by the system during the single-talk periods when there is no near-end signal. ERLE is defined as:

$$ERLE(\text{dB}) = 10\log_{10}\frac{E\{d^2(t)\}}{E\{\hat{s}^2(t)\}} \tag{9}$$

where E is the statistical expectation operation which is realized by averaging. To evaluate the performance of the system during the double-talk periods, the perceptual evaluation of speech quality (PESQ) is used. The PESQ is calculated by comparing the estimated near-end speech against the ground-truth near-end speech during the double-talk periods only. The PESQ score ranges from -0.5 to 4.5 and a higher score indicates a better quality.

4.2. Dataset

We used TIMIT dataset [16] to evaluate the AEC performance. To create the dataset, we followed the steps reported in [8]: From 630 speakers of TIMIT, 200 speakers are randomly chosen to be used as pairs of the far-end and near-end speakers (40 male-female, 30 male-male, 30 female-female). Three utterances of the same far-end speaker are randomly chosen and concatenated to create a far-end signal. Each utterance of a near-end speaker is then extended to the same size as that of the far-end signal by filling zeroes both in front and in rear. Seven utterances of the near-end speakers are used to generate 3500 training mixtures where each near-end signal is mixed with five different far-end signals. From the remaining 430 speakers, we randomly picked another 100 pairs of speakers as the far-end and near-end speakers. We followed the same procedure as described above, but this time only three utterances of the near-end speakers are used to generate 300 testing mixtures where each near-end signal is mixed with one far-end signal. Therefore, the testing mixtures are from untrained speakers.

We used two linear and nonlinear models to simulate the acoustic path. For the nonlinear model of acoustic path [17], we used hard clipping and sigmoidal function to simulate the power amplifier and distortion of loudspeaker respectively as follow:

$$x_{clip}(t) = \begin{cases} -x_{max} \; if \; x(t) < -x_{max} \\ x(t) \quad if \; |x(t)| \leq x_{max} \\ x_{max} \quad if \; x(t) > x_{max} \end{cases}. \tag{10}$$

$$x_{nl}(t) = 4\left(\frac{2}{1+\exp(-a.b(t))} - 1\right) \tag{11}$$

where $b(t) = 1.5x_{clip}(t) - 0.3x_{clip}(t)^2$ and $a = 4$ if $b(t) > 0$ and $a = 0.5$ otherwise. Finally, the linear and nonlinear models of acoustic path are obtained by convolving $x(t)$ and $x_{nl}(t)$with a randomly chosen RIR $g(t)$ as:

$$y_{lin}(t) = x(t) * g(t) \tag{12}$$

$$y_{nl}(t) = x_{nl}(t) * g(t) \tag{13}$$

where $*$ indicates convolution.

The real measured RIRs are selected from the Aachen impulse response database [18]. These RIRs were captured using a mock-up phone in the usual Hand-Held Position (HHP). To generate the training mixtures, we used either synthetic RIRs that are generated using the image method [19] or the real measured RIRs. For the testing mixtures, we used the real measured “corridor” RIR.

For the training mixtures, we generated the microphone signals at a signal to echo ratio (SER) level randomly chosen from {-6, -3, 0, 3, 6}dB by mixing the near-end speech signal and the echo signal. The SER is calculated on the double-talk period as:

$$\text{SER(dB)} = 10\log_{10}\frac{E\{s^2(t)\}}{E\{\text{y}^2(t)\}}. \tag{14}$$

For the testing mixtures, we generated the microphone signals at three different SER levels (0dB, 3.5dB, and 7dB). The unprocessed PESQ scores are calculated by comparing the microphone signal against the near-end signal during the double-talk period.

4.3. Model Architecture Details

In our evaluation, the speech signals are sampled at 16 kHz. The spectral feature vectors are computed using a 512-point STFT with a frame shift of 256-point (16ms). The 512-point STFT magnitude vector is reduced to 257-point by removing the conjugate symmetric half. The final logarithmic magnitude spectral feature vector is extracted by applying the logarithmic operation to the STFT magnitude.

For the FDNLMS updating rule, we set $\mu = 0.5$ when there is no double-talk and we set it to a very small value otherwise. The forgetting factor α in the FDNLMS is set to 0.6. All models were trained using the AMSGrad optimization [20] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-3}$ for 100 epochs. The batch is set to 100. The weights of all layers are initialized with the Xavier method [21] and the biases are set to zero. We set the learning rate to 0.0003. To avoid overfitting, we used the L2 regularization for all the weights with a regularization constant of 0.000001. The input features are normalized to have a mean of zero and a standard deviation of one (unit variance) using the scalars calculated from the training data.

4.4. Results

As the classical benchmark system, we used a FDNLMS method based on [22] with a double-talk detection (DTD) based on the energy of the microphone and far-end signals. We also compared our results against the DNN method presented in [6]. In our implementation of “FDNLMS + DNN”, the parameters of DNN are set to the values given in [6].

Table 1. ERLE and PESQ scores in the linear model of acoustic path when trained on the real measured RIRs with the same device.

	Method	Testing SER (dB)		
		0	3.5	7
ERLE	FDNLMS (Classical)	12.16	11.46	10.52
	FDNLMS + DNN [6]	32.07	32.55	33.67
	E-D GRU	55.12	58.80	60.23
	E-D GRU + CAM	55.54	59.36	61.72
	CAD-AEC	**56.51**	**60.49**	**61.39**
PESQ	Unprocessed	1.86	2.10	2.33
	FDNLMS (Classical)	2.43	2.63	2.81
	FDNLMS + DNN [6]	2.55	2.74	2.90
	E-D GRU	2.74	2.96	3.15
	E-D GRU + CAM	2.88	3.10	3.28
	CAD-AEC	**2.97**	**3.16**	**3.33**

We first evaluated our CAD-AEC in the linear model of acoustic path. In this set of experiments, we used the real measured RIRs captured in “office”, “meeting room”, “lecture room”, “stairway1”, “stairway2”, “bathroom”, and “lecture room” for training and “corridor” for testing in HHP. Here, the mismatch between the training RIRs and the testing RIR is small as the recording device was the same. We calculated the average normalized cross correlation (NCC) between the training and testing RIRs to measure their similarities. The NCC for this case is 0.97. Table 1 shows the average ERLE values and PESQ scores for the classical benchmark, and our proposed CAD-AEC. This table also shows the results for encoder-decoder GRU networks with and without attention that only takes $\log|\boldsymbol{X}|$ and $\log|\boldsymbol{D}|$ as inputs which are denoted as “E-D GRU” and “E-D GRU + CAM” respectively. Even our CAD-AEC method outperforms all other methods, but the performance margin with “E-D GRU + CAM” method is small as the training and testing conditions are very similar.

Table 2. ERLE and PESQ scores in the linear model of acoustic path when trained on synthetic RIRs.

	Method	Testing SER (dB)		
		0	3.5	7
ERLE	E-D GRU + CAM	17.13	20.96	29.68
	CAD-AEC	**42.66**	**47.96**	**52.47**
PESQ	E-D GRU + CAM	2.43	2.68	2.89
	CAD-AEC	**2.76**	**2.92**	**3.06**

We further evaluated the performance of our proposed method when the training and testing conditions are more different than the previous experiments. For this, we generated seven synthetic RIRs for training and again tested on data that was created by the real measured “corridor” RIR. We matched the “corridor” environment per description provided in [18] with reverberation time (T_{60}) selected from {0.2, 0.4, 0.6, 0.8, 0.9, 1.0, 1.25}s. Here the average NCC between the training and testing RIRs is about 0.58. The comparison results are given in Table 2. In this experiments, our CAD-AEC method outperforms the “E-D GRU + CAM” method by a large margin. Fig. 3 illustrates the spectrograms of an AEC example in the linear model of acoustic path with 0dB SER and the real measured RIR for the classical FDNLMS, “E-D GRU + CAM”, and “CAD-AEC” methods when the models are trained using synthetic RIRs. Evidently, the classical method cannot remove the echo signal in single-talk period and recover the near-end speech in double-talk section. Although the “E-D GRU + CAM” method can significantly remove the echo component in this example, there still exist residual echoes (in the marked rectangular). On the other hand, the CAD-AEC method can remove the echoes properly and the near-end speech components are almost completely restored.

We also studied the impact of the nonlinear model of acoustic path on our proposed method. In this set of experiments, we used $y_{nl}(t)$ in generating the microphone signals, therefore our model contains both power amplifier clipping and loudspeaker distortions. We used the synthetic RIRs for training and the “corridor” RIR for testing. We again compared the results of our method against the classical FDNLMS. We also compared our results against the “E-D GRU + CAM”. The results presented in Table 3 show that the proposed method outperforms the other two methods in both PESQ and ERLE.

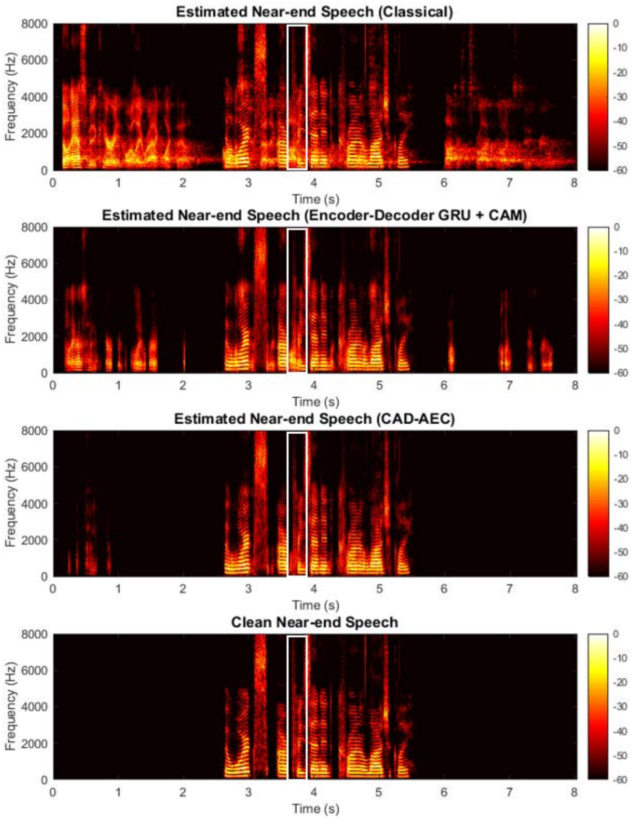


Fig. 3. Spectrograms of the estimated near-end speech for the classical, encoder-decoder GRU with CAM, CAD-AEC, and the clean near-end speech.

Table 3. ERLE and PESQ scores in the nonlinear model of acoustic path when trained on synthetic RIRs.

	Method	Testing SER (dB)		
		0	3.5	7
ERLE	FDNLMS (Classical)	8.16	7.92	7.54
	E-D GRU+ CAM	5.73	5.66	5.91
	CAD-AEC	**19.08**	**19.97**	**21.64**
PESQ	Unprocessed	1.79	2.03	2.27
	FDNLMS (Classical)	2.16	2.39	2.61
	E-D GRU+ CAM	2.12	2.38	2.64
	CAD-AEC	**2.74**	**2.93**	**3.09**

5. CONCLUSION

In this paper, we propose a novel architecture for robust acoustic echo cancellation. To make the trained weights of the proposed model less dependent on the development context, the model is trained with additional adaptive features obtained from frequency domain adaptive filtering of the microphone signal. A contextual attention module is utilized between the deep GRU encoder and decoder to scale the encoder output according to the deployment context. We demonstrate the benefit of integrating FDNLMS with a deep contextual attention AEC over the existing deep learning solutions, especially when there is a great mismatch between the training and testing conditions, where our proposed hybrid AEC network can reduce the echo more significantly while keeping the near-end speech undistorted.

12. REFERENCES

[1] J. Benesty, T. Gansler, D. R. Morgan, S. L. Sondhi, and M. M. Gay, *Advances in Network and Acoustic Echo Cancellation*. Springer, 2001.

[2] S. Haykin, *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[3] S. Gustafsson, R. Martin, and P. Vary, "Combined acoustic echo control and noise reduction for hands-free telephony," *Signal Processing*, vol. 64, no. 1, pp. 21–32, 1998.

[4] D. A. Bendersky, J. W. Stokes, and H. S. Malvar, "Nonlinear residual acoustic echo suppression for high levels of harmonic distortion," in *ICASSP*, 2008, pp. 261–264.

[5] A. Schwarz, C. Hofmann, and W. Kellermann, "Spectral feature-based nonlinear residual echo suppression," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[6] C. M. Lee, J. W. Shin, and N. S. Kim, "DNN-based residual echo suppression," in *INTERSPEECH*, 2015, pp. 1775–1779.

[7] G. Carbajal, R. Serizel, E. Vincent, and É. Humbert, "Multiple-Input Neural Network-Based Residual Echo Suppression," in *ICASSP*, 2018, pp. 231–235.

[8] H. Zhang and D. Wang, "Deep Learning for Acoustic Echo Cancellation in Noisy and Double-Talk Scenarios," in *INTERSPEECH*, 2018, pp. 3239–3243.

[9] H. Zhang, K. Tan, and D. Wang, "Deep Learning for Joint Acoustic Echo and Noise Cancellation with Nonlinear Distortions," in *INTERSPEECH*, 2019, pp. 4255–4259.

[10] A. Fazel, M. El-Khamy, and J. Lee, "Deep Multitask Acoustic Echo Cancellation," in *INTERSPEECH*, 2019, pp. 4250–4254.

[11] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS Workshop on Deep Learning*, 2014.

[13] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv Preprint arXiv:1607.06450*, 2016.

[16] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," in *Workshop on Speech Input/Output Assessment and Speech Databases*, 1989.

[17] S. Malik and G. Enzner, "State-Space Frequency-Domain Adaptive Filtering for Nonlinear Acoustic Echo Cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2065–2079, 2012.

[18] M. Jeub, M. Schäfer, H. Krüger, C. Nelke, C. Beaugeant, and P. Vary, "Do we need dereverberation for hand-held telephony?," in *Proceedings of 20th International Congress on Acoustics*, 2010, pp. 1–7.

[19] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Acoustical Society of America Journal*, vol. 65, pp. 943–950, Apr. 1979.

[20] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," in *ICLR*, 2018.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, vol. 9, pp. 249–256.

[22] C. Faller and J. Chen, "Suppressing acoustic echo in a spectral envelope space," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1048–1061, 2005.