ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413585

A NEURAL ACOUSTIC ECHO CANCELLER OPTIMIZED USING AN AUTOMATIC SPEECH RECOGNIZER AND LARGE SCALE SYNTHETIC DATA

Nathan Howard, Alex Park*, Turaj Zakizadeh Shabestary, Alexander Gruenstein, Rohit Prabhavalkar*

Google Inc., Mountain View, CA, USA

ABSTRACT

We consider the problem of recognizing speech utterances spoken to a device which is generating a known sound waveform; for example, recognizing queries issued to a digital assistant which is generating responses to previous user inputs. Previous work has proposed building acoustic echo cancellation (AEC) models for this task that optimize speech enhancement metrics using both neural network as well as signal processing approaches.

Since our goal is to recognize the input speech, we consider enhancements which improve word error rates (WERs) when the predicted speech signal is passed to an automatic speech recognition (ASR) model. First, we augment the loss function with a term that produces outputs useful to a pre-trained ASR model and show that this augmented loss function improves WER metrics. Second, we demonstrate that augmenting our training dataset of real world examples with a large synthetic dataset improves performance. Crucially, applying SpecAugment style masks to the reference channel during training aids the model in adapting from synthetic to real domains. In experimental evaluations, we find the proposed approaches improve performance, on average, by 57% over a signal processing baseline and 45% over the neural AEC model without the proposed changes.

***Index Terms*—** Acoustic echo cancellation, deep learning, sequence-to-sequence model, multi-task loss, acoustic simulation

1. INTRODUCTION

Voice queries have become increasingly common as a way to communicate with smart devices, such as phones and speakers. In challenging acoustic conditions (background noise, distance from microphone, etc.), interpretation of queries can fail due to poor speech recognition accuracy. We focus on the problem of acoustic echo cancellation (AEC) – removing a known source of additive interference. The term ”echo” cancellation is used because the device has access to the original **reference** signal that is the source of interference, but the interference itself is an echoic version of the signal that has propagated through the room before being received at the microphone(s). In this paper, we will denote the user's speech as the **target** and the mixture of reverberant target and background noise as the **residual**. The received mixture of echoed reference and residual is denoted as the **probe** and the AEC outputs an **erased** signal.

Our end goals are slightly different from typical AEC scenarios because our deployment scenario is echo cancellation in the context of interaction with a smart speaker. As such, there are two important characteristics of the target signal. First, we assume that we are attempting to recover a speech signal – usually a user query. Second, unlike in telephony or meeting situations, the perceptual fidelity of the recovered signal is not as important as its intelligibility to the ASR system. With these considerations in mind, we propose a model and training protocol designed to simultaneously perform echo cancellation, dereverberation, and moderate denoising by learning to predict the target signal given the probe and reference signals.

The contributions of this work are as follows: We propose an autoregressive sequence-to-sequence model for performing acoustic echo cancellation. We demonstrate the value of optimizing on an ASR encoder loss criterion for producing erased signals which improve intelligibility on ASR systems over purely signal-based metrics. Finally we implement two methods for improving robustness of the model to distortion between echo and reference: by preparing a mixture of synthetic and quasi-synthetic data for training, and performing dynamic corruption of the input signals via different configurations of SpecAugment [1].

2. RELATED WORK

In traditional signal processing, linear AEC techniques attempt to estimate the overall system of render-propagation-capture by a time-varying linear filter, usually an adaptive Finite Impulse Response (FIR) filter. Often the filter coefficients are estimated to replicate the echo, in the Minimum Mean Square Error (MMSE) sense, given the reference signal. Then, the filtered version of the reference signal is subtracted from the probe to obtain an estimate of the target signal.

In recent years, there have been numerous proposed approaches to applying neural networks for AEC [2, 3, 4]. In most previous work, the criteria for evaluating AEC performance have been signal driven metrics such as signal distortion ratio (SDR), or echo return loss (ERL). Work here often predicts the residual signal by predicting an ideal ratio mask (IRM) that is applied to the probe [3] or gains applied to the output of a linear AEC [5]. While these metrics are easy to calculate and correlate well to perceptual cancellation quality, our initial experiments indicated that improvements in signal-based metrics often did not translate to proportionally improved WER performance.

Two notable sequence-to-sequence speech prediction models that have been proposed recently are Parrotron [6] and Textual Echo Cancellation [7]. The authors in [6] use an ASR encoder and a text-to-speech (TTS) decoder to perform speech transformation. In order to optimize for intelligibility, the Parrotron model is trained to simultaneously minimize an ASR decoding loss as well as a spectral decoding loss on the same encoded representation. A drawback of this approach is that the transcription of the source signal is required in order to compute the ASR related loss. In [7], the authors assume that the echoed reference is generated by text-to-speech (TTS) and use a Parrotron-style network to remove the echoed reference using only the textual source of the reference signal. The model in that paper uses only spectral loss for training.

*equal contribution

978-1-7281-7605-5/21/$31.00 ©2021 IEEE

3. MODEL ARCHITECTURE

The proposed neural AEC model uses an encoder-decoder structure to reconstruct spectral frames of the erased signal by casting the problem as a sequence-to-sequence task. As in [6] and [7], we use frame level features (80-dimensional log-mel spectral vectors) for both source and target sequences. The source sequence features are computed from the probe and reference signals, and the target sequence features are computed from the clean target signal. Although all three signals should be synchronous, in this system we align probes and references using their cross correlation and enforce that source and target sequences have matching lengths.

The model is comprised of a speech encoder followed by a spectral decoder, which are described in the following sections.

3.1. Encoder

The speech encoder is similar to the encoder described in [8], which takes a sequence of speech features as input and produces a high dimensional hidden representation sequence. We compute feature frames for each of the probe and reference signals, then stack each frame depthwise to create an input tensor that has shape $[B, T, 80, 2]$, where B is the batch size and T is the number of frames. For the encoder used in this work, we used 3 unidirectional LSTM layers, each with 512 hidden dimensions, and no temporal downsampling, so the number of hidden representation has the same number of frames as the input.

3.2. Decoder

We use an autoregressive spectral decoder to predict a sequence of spectral frames from the encoded sequence. The decoder is based on the decoder component described in [9], which is designed to produce spectrogram frames. For the decoder used in this work, we made two small changes. First, because the context needed to transform input to output is local to the frame being processed, we omitted the attention layer. Also, since we constrain the output to be the same number of frames as the input, we also omit the end-of-sequence prediction component of the decoder – the decoder stops producing input when the input frames are exhausted.

The spectral decoder consists of a single 512 dimension LSTM layer followed by an 80 dimension projection layer that feeds its output to a pre-net and a post-net. The pre-net is a feed-forward network that serves to gate the influence of the previous time-step's output compared to the source. The post-net is a stack of five convolutional layers that act on the predicted spectral frames to produce a residual correction factor that is added to create the final prediction. Each non-final convolutional layer applies a 1-dimensional convolution in time, with 512 filters of size 5, followed by batch normalization and tanh activation. The output of the decoder is a sequence of 80-dimensional log-mel spectral frames which can be inverted back to a time domain waveform via Griffin-Lim [10] or by using a neural vocoder [11]. For this paper, we used Griffin-Lim to produce waveform inputs when needed (e.g. for performing recognition evaluations on AEC output).

3.3. Loss Function

Our initial experiments used purely spectral loss when training the network. This loss is computed by summing the mean $L1$ and mean $L2$ (or MSE) distance between the target spectral features and the output of the decoder, both before and after applying the post-net correction.

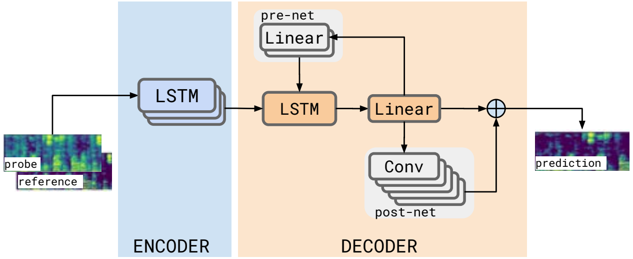

Fig. 1. Model block diagram for inference pathway. The encoder, structured as a typical speech encoder, takes in frame level features and produces a latent representation that is decoded by a spectral decoder to produce frame level features.

Since the motivation for our work is improving speech recognition accuracy on the erased signal predicted by our AEC, we also explored changing the loss function to bias the AEC towards producing outputs useful to an ASR as input. In the ideal case, both the AEC output and target signal would produce the same latent representation when run through the ASR model's encoder. Observing this, we integrated an ASR loss which runs the predicted and target features through an ASR encoder, pre-trained on clean audio, and computes the MSE between the respective latent representations. The final loss term is then

$$\texttt{loss} = \texttt{loss}_{spectral} + \lambda\,\texttt{loss}_{ASR}$$

where λ is a hyperparameter. Figure 2 illustrates how the losses are computed and combined. Unlike the auxiliary decoder loss used by Parrotron, the ASR encoder loss compares latent representations of the predicted and target signals without needing the underlying transcription, which removes the need for labeled training data.

3.4. Training

All of our models are trained using Lingvo [12], which is built on top of TensorFlow [13]. The AEC models were trained with a batch size of 128 using the ADAM optimizer [14] and scheduled sampling [15] on a randomly selected half of the autoregressive decoder input.

The ASR model used in the loss function is a ContextNet [16] CNN-RNN transducer, trained on LibriSpeech to 3.8 WER on `test-clean`. The whole model has 31 million parameters and the encoder contains 23 stacked convolution blocks. Importantly, the AEC model fails to converge when the ASR loss is included at the start of training. We resolve this by making λ dependent on the current training step and linearly ramping λ up from zero to its final value, 0.01, over the first 20k training steps.

Because SpecAugment has been shown to be a useful method of data augmentation for improving WER performance [1, 6], we also experimented with applying SpecAugment to AEC model inputs during training. When using SpecAugment, we masked up to 27 of 80 frequency bins divided between 2 frequency masks and up to 5% of frames split between 10 time masks. Models using SpecAugment trained for 200k steps and models without for 90k steps.

4. DATA PREPARATION

A key challenge in building a neural network based AEC is data collection. In real world recordings, the echoed reference component of the probe can be distorted by non-linearities in the loudspeaker's reproduction of the signal [17]. These distortions can vary at different volumes, temperatures and between different loudspeakers. A common practice is to apply a functional non-linearity to mimic loudspeaker distortion as in [18]. Of course, the highest fidelity way to capture these effects in training data is to record echoed reference outputs in real rooms, but this has the considerable downside of being expensive and time consuming. We used a multi-pronged approach to creating diverse AEC training data - by processing with a room simulator, by combining re-recorded real world data with a room simulator, and by dynamically augmenting the data during training using SpecAugment [1].

4.1. Source Data

For training and evaluation of the AEC techniques compared in this paper, we drew from two sources of speech data: parts of the LibriSpeech corpus were used as both targets and references, and an internal set of TacoTron-generated [9] TTS utterances were used as references. For simulating room environments, we used the room simulator described in [19]. Separately to the echoed reference, background noise was added as described in [19], with noise sources drawn from a set of daily life and cafe noise recordings.

4.2. Training Data

4.2.1. Synthetic Echo

In this setup, the return path of the echoed reference was wholly simulated. The target and reference signals were randomly selected from the `train-clean` portion of the LibriSpeech corpus. For each synthetically noisified utterance, a room configuration was sampled from one of 100,000 possibilities, and the simulated probe, echoed reference, and residuals were computed via simulation. The room configurations were constrained to replicate the geometry of a smart speaker; with the loudspeaker set up as a noise source in a fixed position relative to the microphones. The target source was randomly positioned away from the microphones, with elevation angle restricted to the interval $[45°, 135°]$ and distance varying between 0.25 and 8 meters, with a mean of 2.5 meters. For this simulated condition, the target-to-noise-ratio and target-to-echoed-reference-ratio were randomly chosen in the ranges of (0, 20) and (-20, 0) dB, respectively. In total, there were approximately 153k training utterances produced using the synthetic echo setup.

4.2.2. Re-recorded Echo

In order to account for real loudspeaker-induced distortions and differences between synthetic room impulse responses and real-world room return path effects, we also created a set of re-recorded echo utterances. Drawing from the TTS utterances, we collected re-recorded versions of these utterances as echoed reference signals by playing them out of smart speakers in various conference room environments and recording the resulting output on the smart speaker microphones. 7592 training pairs and 1546 test pairs of (reference, echoed reference) signals were collected in this manner.

The re-recordings were then combined with target signals drawn, without re-recording, from the `train-clean` portion of LibriSpeech using the same room configurations as in Section 4.2.1. The re-recorded echo signal was used directly as the echoed reference, without propagating through the room simulation. Otherwise, the echoed reference, background noise source and simulated path of the target signal were mixed together with the same distribution of SNRs as in Section 4.2.1. Approximately 34k training utterances were produced using this re-recorded setup.

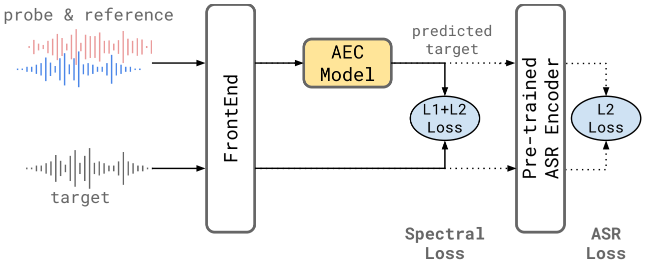

Fig. 2. Two types of losses are are used to optimize the AEC model. Spectral loss is computed between the predicted output and ground truth target features. ASR loss is computed between the encoded representations of the predicted features after passing through a pre-trained ASR encoder. ASR encoder weights are kept fixed while training the AEC model.

4.3. Evaluation data

The test sets for evaluation were constructed as described in Section 4.2.2, but with the test pairs of re-recorded reference signals, target signals drawn from LibriSpeech `test-clean`, and target-to-echoed-reference-ratio levels held fixed at 0dB, -5 dB, and -10 dB to create three test set variants of escalating difficulty. The room impulse responses and background noise samples used for the test sets were all unseen during training.

5. EXPERIMENTS

Unless otherwise specified, speech recognition results were obtained using the ContextNet ASR model described in Section 3.4.

5.1. Data Augmentation Effects

We varied the inputs used for training to gain insight into the effects of augmenting datasets and inputs. These results are shown in Table 1. When controlling for dataset and loss function, we found that applying SpecAugment to the reference signal alone resulted in the most consistent WER reductions. This was the SpecAugment configuration used in our final model during training. Our interpretation of this outcome is that SpecAugment introduces a challenging form of mismatch between the reference and its echo for which the model must compensate and that this mismatch is different from and complementary to the diversity of echoic effects presented by the synthetic/re-recorded data alone.

By looking at matched SpecAugment configurations in Table 1, we observe the benefit of including synthetic *and* re-recorded data in training. Although the re-recorded training data is closest to test set conditions, there are still significant gains from adding training set diversity. When not applying SpecAugment, there was a relative WER reductions of 25.5% (averaged across SNR levels) when using the larger combined dataset compared to the re-recorded data alone.

5.2. ASR Loss Robustness

One concern with optimizing the AEC model using a pretrained ASR encoder is that the AEC will overfit to the idiosyncrasies of that specific ASR encoder and produce outputs that are mismatched when

Training Dataset	SpecAugment	0dB	-5dB	-10dB
Synthetic Reference	None	59.14	70.33	80.75
	Both Inputs	47.68	59.26	71.14
	Probe Only	53.94	66.15	78.01
	Reference Only	31.47	43.23	57.50
Re-recorded Reference	None	19.66	25.35	34.28
Synthetic + Re-recorded	None	14.19	18.71	26.54
	Both Inputs	13.13	15.97	21.98
	Probe Only	12.79	15.83	22.15
	Reference Only	12.51	15.54	22.30

Table 1. Input data effects. WERs of models trained with different SpecAugment configurations and dataset partitions. All models here were trained with **spectral loss only**.

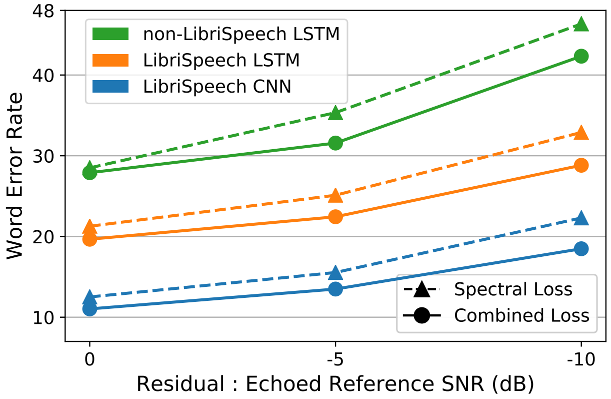

Fig. 3. ASR Loss Robustness. Word Error rates for different ASR models on AEC model outputs, comparing outcomes when training with and without $\texttt{loss}_{ASR}$. Only the LibriSpeech CNN model (blue line) was used for computing $\texttt{loss}_{ASR}$ during training.

presented to other ASR models. To measure this effect, we trained two AEC models, one with and one without $\texttt{loss}_{ASR}$, and evaluated the outputs on three different ASR models, of which only **one** was used for calculating $\texttt{loss}_{ASR}$ during training.

The three pre-trained ASR models evaluated are: a CNN-based global context model [16], a bidirectional LSTM-based listen-attend-spell model [20], and a streaming LSTM-based RNN-T model [21]. The first two models were trained on the train partition of LibriSpeech, and the last model was trained on a large corpus of far field and near field non-LibriSpeech utterances.

Figure 3 shows the WER of each of the ASR models on the outputs of the AEC models. Though only the in-domain CNN speech encoder was used to calculate $\texttt{loss}_{ASR}$ for the AEC model incorporating that loss, we observe consistent improvements for the other two ASR models across all SNRs as well. This improvement holds despite significant differences in model structure, training data, and frontend configuration. As expected, we observe the largest improvements for the matched ASR encoder (CNN), followed by the in-domain LSTM recognizer, and smaller, but still significant gains for the out-of-domain model.

	0 dB	-5 dB	-10 dB
Target*	— 3.81 —		
Residual*	12.18	11.14	12.30
Probe	75.86	82.20	85.78
STFT-based AEC	31.37	32.55	36.91
IRM AEC [3]	23.01	30.75	41.85
Neural AEC (ours)	**11.03**	**13.49**	**18.48**
-AsrLoss	12.51	15.54	22.30
-SpecAugment	14.19	18.71	26.54
-Synthetic Dataset	19.66	25.35	34.28

Table 2. System comparison. WER calculated on different SNR test subsets using various AEC models and available signals. Target and residual are oracle signals not available to the model at inference.

5.3. Final System

We combined all of the proposed modifications to the model and evaluated WER results in Table 2. For comparison purposes, we contrast against two other AEC techniques. The first is a linear AEC system that performs adaptive filtering on STFT subbands, similar to [22], but using longer STFT frames and within-band only filter taps. We also implemented and trained a mask-based neural network AEC model as described in [3]. That model is trained to predict an ideal ratio mask (IRM) that is then used to mask the spectral magnitude of the probe, which is then inverted back to the time domain. During training, the IRM target is computed using the residual and echoed reference. When training this model, we used both synthetic and re-recorded data, but did not apply SpecAugment.

As expected, all AEC techniques significantly improve recognition accuracy compared to evaluating on the probe signal alone. Moreover, both neural models improve over the STFT-based AEC at higher SNRs (0 dB and -5 dB), but the IRM-based model degrades much more sharply than the STFT-based AEC as SNR decreases. Our neural model, when including all proposed improvements, achieves significant improvements compared to both alternatives at all three SNR levels. In addition to the analysis in Sections 5.1 and 5.2, we show ablation results from successively removing each of the proposed improvements from the final system.

Interestingly, our final model produces outputs that yield better WER in the 0 dB case than running recognition on the residual signal directly, which has no echoed reference. This is presumably because the model was trained with non-reverberant, noise-free utterances as its training targets and therefore learned to predict de-reverberated and de-noised features rather than just the residual.

6. CONCLUSION

We proposed an autoregressive neural network model to perform AEC in situations with double-talk and background noise. The model was trained using a dataset augmented with synthetic examples with SpecAugment masks applied to increase robustness to mismatch between the reference and the echoed reference. To adapt the model towards being an input to an ASR system, the loss function was extended with a pretrained ASR encoder. When compared to a purely signal processing-based AEC technique and a mask-based neural AEC model, our proposed approach improved speech recognition accuracy across several noise levels.

Acknowledgements: Thanks to James Walker and Bharath Ranganatha Mankalale for aid in data collection and room simulation.

7. REFERENCES

[1] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[2] Qinhui Lei, Hang Chen, Junfeng Hou, Liang Chen, and Lirong Dai, "Deep Neural Network Based Regression Approach for Acoustic Echo Cancellation," in *Proceedings of ICMSSP 2019*, 05 2019, pp. 94–98.

[3] Hao Zhang and DeLiang Wang, "Deep Learning for Acoustic Echo Cancellation in Noisy and Double-Talk Scenarios," in *Proceedings of Interspeech*, 09 2018, pp. 3239–3243.

[4] Amin Fazel, Mostafa El-Khamy, and Jungwon Lee, "Deep Multitask Acoustic Echo Cancellation," in *Proceedings of Interspeech*, 09 2019, pp. 4250–4254.

[5] Chul Min Lee, Jong Won Shin, and Nam Soo Kim, "DNN-based residual echo suppression," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] Fadi Biadsy, Ron J. Weiss, Pedro J. Moreno, Dimitri Kanevsky, and Ye Jia, "Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-Impaired Speech and Speech Separation," *arXiv preprint arXiv:1904.04169*, 2019.

[7] Shaojin Ding, Ye Jia, Ke Hu, and Quan Wang, "Textual Echo Cancellation," *arXiv preprint arXiv:2008.06006*, 2020.

[8] Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, et al., "Streaming End-to-end Speech Recognition For Mobile Devices," *CoRR*, vol. abs/1811.06621, 2018.

[9] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017.

[10] D. Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[11] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, et al., "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[12] Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia Xu Chen, et al., "Lingvo: a Modular and Scalable Framework for Sequence-to-Sequence Modeling," *CoRR*, vol. abs/1902.08295, 2019.

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *CoRR*, vol. abs/1603.04467, 2016.

[14] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2017.

[15] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[16] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, et al., "ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context," *arXiv preprint arXiv:2005.03191*, 2020.

[17] Gerald Enzner, Herbert Buchner, Alexis Favrot, and Fabian Kuech, "Chapter 30. Acoustic Echo Control," *Academic Press Library in Signal Processing*, vol. 4, 12 2014.

[18] Hongsheng Chen, Teng Xiang, Kai Chen, and Jing Lu, "Non-linear Residual Echo Suppression Based on Multi-stream Conv-TasNet," *arXiv preprint arXiv:2005.07631*, 2020.

[19] Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes, Arun Narayanan, et al., "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Proceedings of Interspeech*, 2017, pp. 379–383.

[20] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, Attend and Spell," *arXiv preprint arXiv:1508.01211*, 2015.

[21] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, et al., "A Streaming On-Device End-To-End Model Surpassing Server-Side Conventional Model Quality and Latency," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6059–6063.

[22] Y. Avargel and I. Cohen, "Performance analysis of cross-band adaptation for subband acoustic echo cancellation," in *International Workshop on Acoustic Echo and Noise Control*, 09 2006.