DeepVQE: Real Time Deep Voice Quality Enhancement for Joint Acoustic Echo Cancellation, Noise Suppression and Dereverberation

Evgenii Indenbom, Nicolae-Cătălin Ristea, Ando Saabas, Tanel Pärnamaa, Jegor Gužvin, Ross Cutler

Microsoft Corp.

ando.saabas@microsoft.com

Abstract

Acoustic echo cancellation (AEC), noise suppression (NS) and dereverberation (DR) are an integral part of modern full-duplex communication systems. As the demand for teleconferencing systems increases, addressing these tasks is required for an effective and efficient online meeting experience. Most prior research proposes solutions for these tasks separately, combining them with digital signal processing (DSP) based components, resulting in complex pipelines that are often impractical to deploy in real-world applications. This paper proposes a real-time cross-attention deep model, named DeepVQE, based on residual convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to simultaneously address AEC, NS, and DR. We conduct several ablation studies to analyze the contributions of different components of our model to the overall performance. DeepVQE achieves state-of-the-art performance on non-personalized tracks from the ICASSP 2023 Acoustic Echo Cancellation Challenge and ICASSP 2023 Deep Noise Suppression Challenge test sets[1], showing that a single model can handle multiple tasks with excellent performance. Moreover, the model runs in real-time and has been successfully tested for the Microsoft Teams platform.

Index Terms: acoustic echo cancellation, noise suppression, dereverberation, speech enhancement, deep learning, real-time processing

1. Introduction

Teleconferencing systems like Microsoft Teams, Skype, and Zoom have seen a surge in demand due to the rise of remote work in fields such as business, education, and healthcare. To ensure that such systems provide a productive and pleasant experience to users, it is crucial that they provide good call quality. Noise and acoustic echo are among the main causes of call quality degradation that can significantly diminish speech intelligibility and hinder communication [1]. Those problems become even more challenging in full duplex communication when echo interferes with double-talk (DT) scenarios [2]. Therefore, solutions that can address acoustic echo, noise, and dereverberation are essential for enabling seamless communication.

Although acoustic echo, noise, and reverberation are theoretically three separate effects, they are interdependent in real communication systems, which can make it challenging to address them individually. For instance, in high noise or reverberant scenarios, the performance of the AEC system is negatively affected. Most related works have focused on each task separately [3–12]. However, such an approach involves cascading the AEC, NS and DR systems, which leads to more complex communication pipelines. To address this shortcoming, we propose a more natural approach by canceling echoes, noise, and reverb with a joint deep learning model.

Recently, joint AEC and NS [3, 13–15] methods have been developed to simplify the communication pipeline while providing a good AEC and NS performance. For example, MTFAA-Net [3] is a neural network for joint AEC and NS, based on multi-scale time-frequency processing and streaming axial attention. The network provided the best results in 2022 AEC and DNS challenges, showing that the model can handle joint tasks with state-of-the-art (SOTA) performance. Still, MTFAA-Net relies on classical AEC components, such as a signal processing-based linear echo canceller (LAEC) and a delay compensator for aligning the microphone and the far end signals. In fact, in the ICASSP 2022 AEC Challenge [16], the top 4 submissions employed a DSP-based LAEC and a delay compensator for signal alignment [3, 17–19]. We show a non-hybrid approach can give SOTA performance.

The methods [6, 20, 21] perform in-model alignment between the microphone and far end signals, which can replace and improve DSP-based alignment methods [22]. In [20] a model which performs alignment in the time domain with a local attention block is described, where the attention block computes the alignment based on the RNN's internal states. Distinctly, our model performs attention to deep time-frequency features, computing an actual delay distribution which is used to soft align the far end features. More closely to our work, [6] uses a cross-attention mechanism for deep features alignment. We enhance that mechanism by adding a convolutional layer in the time-delay map, which stabilizes the delay distribution and enhances the AEC performance. There has been a wide range of NS methods (e.g., [4, 13, 14]) with remarkable results, but without the ability to jointly perform AEC.

In this paper we describe DeepVQE, a residual CNN autoencoder with a Gated Recurrent Unit [23] (GRU) bottleneck. Considering that one major issue in AEC systems is the delay between the microphone and the far end reference signals [6], our network contains a cross-attention block to soft-align the near end and far end signals. To provide a better output signal quality, we employed a complex convolving mask (CCM) block, being inspired by [24], which allows the network to estimate each time-frequency bin by mixing multiple neighbor bins in a learnable fashion. Moreover, being inspired by the success in the computer vision field, we replaced the standard upscaling blocks from the decoder with sub-pixel convolutional blocks [25], enabling a higher feature diversity with a small performance cost. Our contribution are:

- We developed a new cross-attention mechanism for the microphone and far end soft alignment in feature space.

- We developed a new architecture which efficiently combines

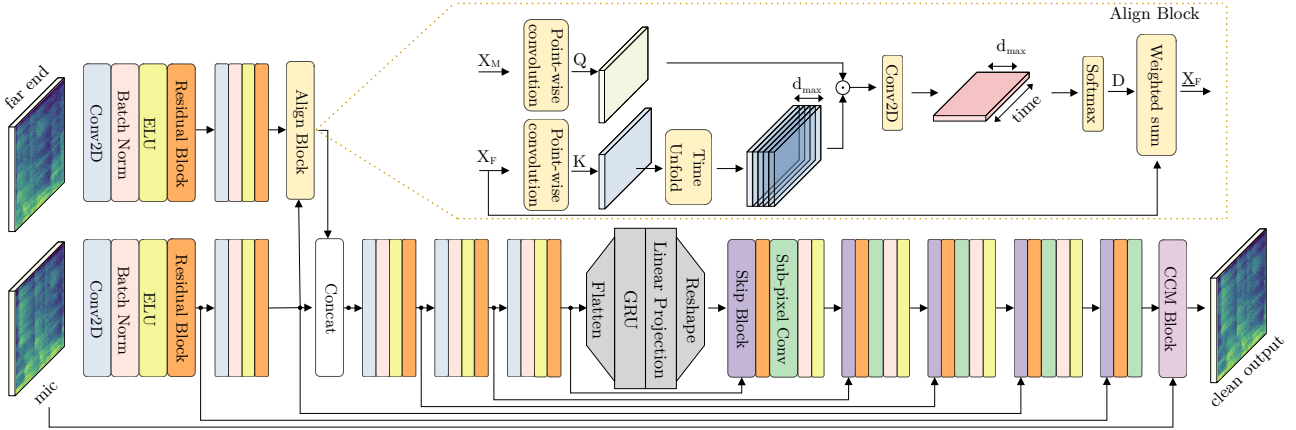[1]This paper was not entered in these ICASSP 2023 challenges

Figure 1: *DeepVQE architecture overview. Using the mic and far end signal, the network reconstructs the clean mic signal, without undesired echoes, noises, and reverb. At the top, the cross-attention-based alignment block is illustrated. Best viewed in color.*

multiple ideas (alignment block, residual block, CCM, sub-pixel convolution) for joint AEC, NS and DR tasks.

- We obtained state-of-the-art performance on both Acoustic Echo Cancellation Challenge 2023 [26] and Deep Noise Suppression Challenge 2023 [27] with our joint model.
- DeepVQE-S joint model is performant enough for real-time workloads on even low end devices and has been successfully tested in Microsoft Teams for hundreds of millions of users.

2. Proposed Method

Problem Formulation. We consider the following generic communication system for AEC, NS, and DR tasks: the far end reference signal is transmitted to the receiving room, played back through the loudspeaker, and then picked up by the microphone via an acoustic echo path (modeled by a room impulse response). The captured microphone signal is composed of near end signal, background noise, reverberations, and echoes. The captured microphone signal is then processed by the speech enhancement component, and the produced clean signal is sent to the far end user. The speech enhancement component's responsibility is to perform echo cancellation, as well as to remove background noises and reverberation from the near end signal. Our goal is to provide a single model performing the joint task of removing undesired echoes, noises, and reverberation, having the microphone and the reference far end signals as inputs.

Feature extraction. Because there is not a significant perceptual difference between fullband (48 kHz) and super wideband (24 kHz) signals [28], we chose to sample input and output audio at 24 kHz to achieve better inference speed. The 48 kHz signals are downsampled, enhanced with our models, and upsampled back to the original sampling rate. The preprocessing is performed identically for the reference far end and microphone signals. The input features to the network are power law compressed complex spectra computed with a squared root Hann window [29].

Overall network architecture. The DeepVQE architecture is depicted in Figure 1. It consists of encoder, GRU bottleneck, decoder and CCM block, which we describe below. In this section we use $c, t, f \in \mathbb{N}$ to denote channel, time, and frequency axis lengths.

Encoder. The encoder is composed of the mic and far end branches. The microphone branch has five encoding blocks, while the far end branch has only two, followed by the align-

ment block. The alignment block aligns far end and microphone features in time. The aligned far end and microphone features are concatenated and fed into the third encoding block in the microphone branch. Each encoding block is built by stacking a downsampling convolutional layer, batch-norm, ELU function, and residual block (see below). The first microphone encoding block has 64 filters and all the following microphone encoding blocks have 128 filters. The far end branch has 32 filters in the first block and 128 filters in the second. The downsampling convolutional layers have kernel size 4×3 and a stride of 1×2, reducing the number of bins along the frequency axis. The convolutions are causal, meaning that the padding is performed so that no look-ahead is used. The encoder is shown in the left part of Figure 1.

Residual block. In both the encoding and decoding blocks we added a residual block, which increases the network's capacity, while not hindering the gradient flow through the network. The block consists of a convolutional layer, followed by batch-norm and exponential linear unit (ELU) [30] activation. The block could be formally defined as:

$$\mathbf{Y} = \mathbf{X} + ELU(BatchNorm(Conv2D(\mathbf{X}))), \quad (1)$$

where $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{c \times t \times f}$ are input and output tensors respectively. The convolution layer in the residual block has the same number of filters as the number of channels in the input, kernel size of 4×3, the stride of 1, and causal padding such as the input shape is preserved.

Alignment block. Let $\mathbf{X}_M \in \mathbb{R}^{c \times t \times f}$ be the mic features and $\mathbf{X}_F \in \mathbb{R}^{c \times t \times f}$ the far end features. The feature maps are processed by point-wise convolution layers into $\mathbf{Q} \in \mathbb{R}^{h \times t \times f}$ and $\mathbf{K} \in \mathbb{R}^{h \times t \times f}$, where h is the number of similarity channels. Next, we unfold on the time axis the $\mathbf{K}$, creating a delay dimension and changing the shape to $\mathbf{K}_u \in \mathbb{R}^{h \times t \times d_{max} \times f}$, where d_{max} is the maximum echo delay expressed in time frames. Afterward, we perform a dot product on the frequency axis between the query and the unfolded key, obtaining $\mathbf{Z} \in \mathbb{R}^{h \times t \times d_{max}}$. The results are fed into a convolutional layer with a kernel size of 5×3, padding of 3×1, and stride 1. The convolution has a single filter combining h similarity channels into a single attention head, which is further processed by a softmax on the delay axis, outputting a delay probability distribution $\mathbf{D} \in \mathbb{R}^{t \times d_{max}}$. Finally, the aligned far end features $\underline{\mathbf{X}}_F \in \mathbb{R}^{c \times t \times f}$ are computed as a weighted sum on the time axis with the corresponding delay probabilities from $\mathbf{D}$. More

precisely, for each delay value in $[0, d_{max})$, $\mathbf{X}_F$ is delayed, multiplied by the corresponding weight factor from $\mathbf{D}$, and added to the final result $\underline{\mathbf{X}}_F$.

Bottleneck. The bottleneck is located between the encoder and decoder and consists of a recurrent layer and a linear projection. The recurrent layer input is feature maps from the encoder flattened along the channel and frequency dimensions. Formally, the input $\mathbf{X} \in \mathbb{R}^{c \times t \times f}$ is flattened into $\mathbf{X} \in \mathbb{R}^{t \times (c \cdot f)}$. Afterward, $\mathbf{X}$ is processed into the recurrent layer, fed into the linear projection, and then the linear projection output is reshaped back to $\mathbf{X} \in \mathbb{R}^{c \times t \times f}$. Following [29], considering that an LSTM does not bring significant performance improvements, we use a GRU layer to reduce the model complexity. Using linear projection after the recurrent layer allows for a reduction of the number of hidden units in the recurrent layer improving both performance and training stability.

Decoder. The decoder consists of five decoding blocks. All but the last one is built by stacking a skip block, residual block, sub-pixel convolution block, batch-norm, and ELU function. The last decoding block consists of a skip block, residual block, and sub-pixel convolution block only. The number of filters in the decoder is changed only in the sub-pixel convolution blocks, while the other blocks preserve the tensor shape. The number of filters for the sub-pixel blocks is 128, 128, 128, 64, and 27. Each sub-pixel convolution has the kernel size of 4×3 and a stride of 1. All convolutions are causal, meaning that the padding is added so that no look-ahead is performed.

Skip block. We replaced the classical skip connection, based on concatenation or summing, with a convolutional layer, having a kernel size of 1×1 and a stride of 1. The encoder features are point-wise projected and then summed with the corresponding decoder output. Besides decoupling the encoder and decoder feature spaces, the point-wise convolution allows us to choose the number of channels in the encoder independently from the number of channels in the decoder and obtain better results for the performance-speed trade-off.

Sub-pixel convolution. After downscaling the input on the frequency axis in the encoder part, we need to upscale back to the original resolution in the decoder. We replaced the regular upscaling method based on transposed convolution with the sub-pixel convolution [25], which learns an array of filters to upscale the low-resolution feature maps into the high-resolution output. Each upscaling is performed with a factor of two on the frequency axis. Formally, the $\mathbf{X} \in \mathbb{R}^{c_i \times t \times f}$, having $c_i \in \mathbb{N}$ channels, is transformed by a regular convolution with $2c$ filters into $\mathbf{X}' \in \mathbb{R}^{2c \times t \times f}$, and then transposed and reshaped into the actual output $\mathbf{Y} \in \mathbb{R}^{c \times t \times 2f}$.

Complex convolving mask block. The complex convolving mask block consists of two stages. The first stage builds the complex-valued mask by splitting the input channels into three weight components. Each component is the weight of a 120 degree rotating vector in the complex plane. We define $\mathbf{v} = (v_1, v_2, v_3) = (1, -\frac{1}{2} + j\frac{\sqrt{3}}{2}, -\frac{1}{2} - j\frac{\sqrt{3}}{2})$ and reshape the input $\mathbf{X} \in \mathbb{R}^{c \times t \times f}$ into $\mathbf{X}' \in \mathbb{R}^{3 \times \frac{c}{3} \times t \times f}$. Next, we compute the complex mask $\mathbf{H} \in \mathbb{C}^{\frac{c}{3} \times t \times f}$ as described in Equation 2.

$$\mathbf{H} = \mathbf{v} \cdot \mathbf{X}' \qquad (2)$$

Considering that the angle between $\mathbf{v}$ components is 120 degrees in the complex space, the complex mask covers the entire complex plane. In practice we observed that using a three-vector component instead of the regular two-vector component (real and imaginary parts) offers more stable output results, preventing low noise and echo leakage.

In the second stage, we reshape the channel dimension of the complex mask $\mathbf{H}$ to form a $(m + 1) \times (2n + 1)$ convolution kernel $\mathbf{M} \in \mathbb{C}^{(m+1) \times (2n+1) \times \frac{c}{3(m+1)(2n+1)} \times t \times f}$ with weights varying over time and frequency dimensions. But, as the input microphone spectrum is a single channel complex valued tensor $\mathbf{X}_{mic} \in \mathbb{C}^{t \times f}$, we need to enforce $c = 3(m + 1)(2n + 1)$. Therefore, having the input microphone spectrum $\mathbf{X}_{mic} \in \mathbb{C}^{t \times f}$ and the complex convolving mask $\mathbf{M} \in \mathbb{C}^{(m+1) \times (2n+1) \times t \times f}$ (after squeezing the redundant channel dimension), the clean spectrum $\hat{\mathbf{X}} \in \mathbb{C}^{t \times f}$ is estimated as described in Equation 3. The input spectrum $\mathbf{X}_{mic}$ is padded with zeros to ensure that clean spectrum $\hat{\mathbf{X}}$ is produced for all frames and frequency bins.

$$\hat{\mathbf{X}}(t, f) = \sum_{i=-m}^{0} \sum_{j=-n}^{n} \mathbf{X}(t + i, f + j) \cdot \mathbf{M}(i, j, t, f) \qquad (3)$$

Computing a deep filter for output reconstruction helps the network to leverage neighbor time-frequency bins in a learnable fashion. The CCM block is applied causally.

3. Experiments

Datasets. To ensure the generalization capacity, the training data are synthesized online from clean and noisy speech, with random parameters for each sample (e.g., signal-to-noise ratio, room impulse response, distortion, gain, signal-to-echo ratio etc.). We sample clean and noisy speech, and noise recordings for training from data provided in the ICASSP 2022 AEC [16] and DNS [32] challenges.

We report the final results on the blind test sets from the ICASSP 2023 AEC [26] and DNS [27] challenges. As the DeepVQE model processes audio sampled at 24 kHz, we downsample the blind test set audio and upsample the result as described in Section 2.

Experimental setup. To test the echoes removal, we employ the echo return loss enhancement (ERLE) for far end single-talk (FEST) scenarios and AECMOS [31] echo score for both FEST and DT. For near end single-talk (NE) scenarios, we use the word error rate (WER), AECMOS [31] degradation score, DNSMOS P.835 signal (SIG), background (BAK), overall (OVRL) scores [33] and the speech-to-reverberation-ratio (SRR), estimated with an internal model. In the NS scenarios, we feed into the network an empty far end signal.

Hyper-parameters. For feature generation, we used a squared root Hann window of length 20ms, a hop length of 10ms, and a discrete Fourier transform length of 480. This leads to 20ms algorithmic delay, consisting of 10ms output signal delay (due to overlap-add signal reconstruction) and packet (frame) duration (10ms). We trained all the networks using AdamW [34] optimizer with batches of 400 samples for 250 epochs, with a learning rate of $1.2 \cdot 10^{-3}$ and a weight decay of $5 \cdot 10^{-7}$. Similar to [6], we have taken $d_{max} = 100$, which is equivalent to the maximum delay of 1 second.

Ablation study. We performed the ablation studies with the small configuration of our DeepVQE model, named DeepVQE-S. DeepVQE-S is a downscaled version of our best model. The DeepVQE-S microphone branch has 4 blocks with 16, 40, 56, and 24 filters, the far end branch has 8 and 24 filters, and the decoding branch has 4 blocks with 40, 32, 32, and 27 filters. Additionally, the residual block is omitted in all the encoder blocks and in the first and last decoder blocks to save more computing. We find it is more interesting to show ablation results on the production-sized model, where the quality impact of architectural changes is especially important.

Table 1: *Ablation study results for our small DeepVQE-S model on LD-M, LD-H [6], and 2023 AEC challenge [26] blind test set, for far end single-talk (AEC-FEST) and double-talk (AEC-DT) scenarios. We compare the DeepVQE-S model without an alignment block (shown as DSP-aligned), with the alignment block proposed in [6] and with our alignment block (shown as Ours). For the model without the alignment block, the far end and mic signals are aligned using the DSP-based method. We report the ERLE, WER, AECMOS Echo (AECMOS$_e$) and Degradation (AECMOS$_d$) [31].*

Align method	LD-M		LD-H		AEC-FEST		AEC-DT		
	ERLE↑	AECMOS$_e$↑	ERLE↑	AECMOS$_e$↑	ERLE↑	AECMOS$_e$↑	AECMOS$_e$↑	AECMOS$_d$↑	WER↓
DSP-aligned	41.76	4.15	33.18	3.96	54.12	4.45	**4.62**	3.89	36.27
[6]	59.19	4.57	54.49	4.46	61.04	4.56	4.60	3.95	36.23
Ours	**61.22**	**4.60**	**55.51**	**4.49**	**65.70**	**4.61**	**4.62**	**4.02**	**31.79**

Table 2: *Ablation study results for our small DeepVQE-S model on 2023 AEC Challenge [26] near end single-talk (AEC-NEST) blind test set and 2022 DNS Challenge [32] blind test set. The model includes our alignment block although the far end signal is empty in nearend scenario. We report the AECMOS Degradation (AECMOS$_d$) [31], DNSMOS P.835 SIG, BAK, OVRL scores [33] and the SRR, estimated with an internal model. On the first line, we included for comparison the scores for noisy data.*

Residual Block	Sub-Pixel conv	CCM Block	AEC-NEST					DNS 2022				
			AECMOS$_d$↑	SIG↑	BAK↑	OVRL↑	SRR↑	AECMOS$_d$↑	SIG↑	BAK↑	OVRL↑	SRR↑
-	-	-	3.27	3.70	3.08	2.90	25.18	2.72	3.52	2.10	2.28	25.56
	✓	✓	4.35	3.82	4.30	3.52	35.83	4.04	3.57	4.06	3.26	36.71
✓		✓	4.35	3.83	4.32	3.55	35.62	4.07	3.58	4.03	3.26	36.45
✓	✓		4.29	3.80	4.30	3.51	34.65	3.95	3.54	4.02	3.22	35.71
✓	✓	✓	**4.36**	**3.84**	**4.35**	**3.56**	**36.27**	**4.09**	**3.60**	**4.10**	**3.30**	**36.98**

Table 3: *2023 DNS Challenge MOS results [27]. We included the competition baseline and the non-personalized winner.*

	Method	SIG	BAK	OVRL	WAcc	M
Track 1	DNS Baseline	3.14	2.60	2.34	70.7%	0.521
	DNS Winner [27]	**3.58**	2.82	2.65	72.5%	0.569
	DeepVQE	3.47	**2.94**	**2.73**	**73.4%**	**0.582**
Track 2	DNS Baseline	3.22	2.68	2.38	72.7%	0.536
	DNS Winner [27]	**3.64**	2.88	2.66	72.4%	0.570
	DeepVQE	3.57	**3.06**	**2.83**	**76.0%**	**0.608**

Table 4: *2023 AEC Challenge MOS results [26]. We included the competition baseline and the non-personalized winner.*

Method	ST FE Echo	DT Echo	DT Other	ST NE SIG	ST NE BAK	WAcc	Final score
AEC Baseline	4.53	4.28	3.47	3.88	3.88	64.9%	0.736
AEC winner	**4.70**	**4.77**	**4.31**	3.99	4.38	**82.3 %**	0.852
DeepVQE-S	4.66	4.63	4.00	4.04	4.33	75.7%	0.821
DeepVQE	4.69	4.70	4.29	**4.15**	**4.41**	80.7%	**0.854**

In Table 1, we compare our alignment block against the DSP-based alignment method (the DeepVQE-S without the alignment block) and the alignment block proposed in [6]. We observe that our method surpasses both methods in each and every metric with a considerably higher improvement obtained for WER. We highlight that all the compared models are derived from the DeepVQE-S model and include residual blocks, sub-pixel convolutions, CCM blocks, etc. An extended ablation for the AEC task with comparative audio samples and visualizations of the alignment delay map is presented in `https://ristea.github.io/deep-vqe`.

In Table 2, we present the ablation results for the NS task on both 2022 DNS Challenge [32] and 2023 AEC Challenge [26] blind test sets. We remove or replace each of the proposed blocks to see the impact of each architectural change separately. The biggest improvement is provided by the CCM block, showing the potential of utilizing magnitude and phase information from neighboring frequency bins and preceding time frames. In terms of DR, our best DeepVQE-S shows over 10dB SRR improvement on both AEC-NEST and DNS data, showing a great capacity to jointly perform AEC, NS and DR.

Challenge results. In Table 3 and Table 4 we included the subjective mean opinion score (MOS), word accuracy rate (WAcc), and the overall final score from the 2023 AEC [26] and DNS [27] challenges for our best DeepVQE model. Considering that our model does not use personalized information, we compared ourselves with the non-personalized models. We significantly overpass the winner of the DNS Challenge for both tracks, obtaining better results for 3 out of 4 metrics. Regarding the AEC Challenge winner, we obtained better metrics for ST NE scenarios, while being slightly behind for ST FE and DT scenarios. Nevertheless, according to the official final score, we rank first.

We highlight that the winners from AEC and DNS challenges are different models specifically designed and trained for the challenge task, while DeepVQE is exactly the same model, trained to jointly perform both tasks.

Inference speed. Providing SOTA performance within a speed-parameters budget is critical to deploy the models in production for teleconferencing applications. DeepVQE has 7.5M parameters and an inference time of 3.66ms per frame on a CPU Intel Core i7 11370H@3.3 GHz, while DeepVQE-S has only $0.59M$ parameters and an inference time of 0.14ms per frame. Having a very good performance and a real-time factor of 0.014, DeepVQE-S has been successfully tested in Microsoft Teams for hundreds of millions of users.

4. Conclusions

In this paper, we propose the DeepVQE architecture, a new model for real-time unified AEC, NS, and DR. Our model contains a more stable alignment block, which significantly improves the AEC performance. Moreover, the model contains residual blocks, sub-pixel convolutions, and CCM blocks, which enables the model to attain SOTA performance in NS. DeepVQE model achieved SOTA performance in both 2023 AEC [26] and 2023 DNS [27] challenges, which leads the way to unified speech modeling. In the future, we aim to improve the performance of the proposed model and extend the architecture for personalized AEC and NS.

5. References

[1] V. Grancharov and W. B. Kleijn, "Speech Quality Assessment," *Springer Handbook of Speech Processing*, pp. 83–100, 2008.

[2] K. Sridhar, R. Cutler, A. Saabas, T. Parnamaa, M. Loide, H. Gamper, S. Braun, R. Aichner, and S. Srinivasan, "ICASSP 2021 Acoustic Echo Cancellation Challenge: Datasets, Testing Framework, and Results," in *Proceedings of ICASSP*. IEEE, 2021, pp. 151–155.

[3] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-Scale Temporal Frequency Convolutional Network with Axial Attention for Speech Enhancement," in *Processing of ICASSP*. IEEE, 2022, pp. 9122–9126.

[4] C. Zheng, X. Peng, Y. Zhang, S. Srinivasan, and Y. Lu, "Interactive Speech and Noise Modeling for Speech Enhancement," in *Proceedings of the AAAI*, vol. 35, no. 16, 2021, pp. 14 549–14 557.

[5] H. Zhang, S. Kandadai, H. Rao, M. Kim, T. Pruthi, and T. Kristjansson, "Deep Adaptive AEC: Hybrid of Deep Learning and Adaptive Acoustic Echo Cancellation," in *Processing of ICASSP*. IEEE, 2022, pp. 756–760.

[6] E. Indenbom, N.-C. Ristea, A. Saabas, T. Pärnamaa, and J. Gužvin, "Deep Model with Built-in Cross-Attention Alignment for Acoustic Echo Cancellation," *arXiv preprint arXiv:2208.11308*, 2022.

[7] Y. Ju, W. Rao, X. Yan, Y. Fu, S. Lv, L. Cheng, Y. Wang, L. Xie, and S. Shang, "TEA-PSE: Tencent-Ethereal-Audio-Lab Personalized Speech Enhancement System for ICASSP 2022 DNS CHALLENGE," in *Processing of ICASSP*. IEEE, 2022, pp. 9291–9295.

[8] S. Zhang, Z. Wang, J. Sun, Y. Fu, B. Tian, Q. Fu, and L. Xie, "Multi-Task Deep Residual Echo Suppression with Echo-Aware Loss," in *Processing of ICASSP*. IEEE, 2022, pp. 9127–9131.

[9] S. Soni, R. N. Yadav, and L. Gupta, "State-Of-The-Art Analysis of Deep Learning-Based Monaural Speech Source Separation Techniques," *IEEE Access*, 2023.

[10] H. Zhang and D. Wang, "A Deep Learning Approach to Multi-Channel and Multi-Microphone Acoustic Echo Cancellation," in *Proceedings of INTERSPEECH*, 2021, pp. 1139–1143.

[11] N. L. Westhausen and B. T. Meyer, "Dual-Signal Transformation LSTM Network for Real-Time Noise Suppression," 2020.

[12] C. Zheng, Y. Zhou, X. Peng, Y. Zhang, and Y. Lu, "Time-Variance Aware Dynamic Kernel Generation for Real-Time Acoustic Echo Cancellation," *IEEE Signal Processing Letters*, vol. 29, pp. 967–971, 2022.

[13] Z. Xu, M. Strake, and T. Fingscheidt, "Deep Noise Suppression Maximizing Non-Differentiable PESQ Mediated by a Non-Intrusive PESQNet," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1572–1585, 2022.

[14] J. Chen, Z. Wang, D. Tuo, Z. Wu, S. Kang, and H. Meng, "FullSubNet+: Channel Attention Fullsubnet with Complex Spectrograms for Speech Enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7857–7861.

[15] Z. Wang, Y. Na, B. Tian, and Q. Fu, "NN3A: Neural Network Supported Acoustic Echo Cancellation, Noise Suppression and Automatic Gain Control for Real-Time Communications," in *Proceedings of ICASSP*. IEEE, 2022, pp. 661–665.

[16] R. Cutler, A. Saabas, T. Parnamaa, M. Purin, H. Gamper, S. Braun, K. Sørensen, and R. Aichner, "ICASSP 2022 Acoustic Echo Cancellation Challenge," in *Processing of ICASSP*. IEEE, 2022, pp. 9107–9111.

[17] H. Zhao, N. Li, R. Han, L. Chen, X. Zheng, C. Zhang, L. Guo, and B. Yu, " A Deep Hierarchical Fusion Network for Fullband Acoustic Echo Cancellation," in *Proceedings of ICASSP*. IEEE, 2022.

[18] S. Zhang, Z. Wang, J. Sun, Y. Fu, B. Tian, Q. Fu, and L. Xie, " Multi-Task Deep Residual Echo Suppression with Echo-aware Loss," in *Proceedings of ICASSP*. IEEE, 2022.

[19] X. Sun, C. Cao, Q. Li, L. Wang, and F. Xiang, " Explore Relative and Context Information with Transformer for Joint Acoustic Echo Cancellation and Speech Enhancement," in *Proceedings of ICASSP*. IEEE, 2022.

[20] L. Ma, S. Yang, Y. Gong, X. Wang, and Z. Wu, "Echofilter: End-to-end Neural Network for Acoustic Echo Cancellation," *arXiv preprint arXiv:2105.14666*, 2021.

[21] Y. Liu, Y. Shi, Y. Li, K. Kalgaonkar, S. Srinivasan, and X. Lei, "SCA: Streaming Cross-attention Alignment for Echo Cancellation," *arXiv preprint arXiv:2211.00589*, 2022.

[22] J. Ianniello, "Time Delay Estimation via Cross-Correlation in the Presence of Large Estimation Errors," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 6, pp. 998–1003, 1982.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *Proceedings of NIPS*, 2014.

[24] W. Mack and E. A. Habets, "Deep Filtering: Signal Extraction and Reconstruction Using Complex Time-Frequency Filters," *IEEE Signal Processing Letters*, vol. 27, pp. 61–65, 2019.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *Proceedings of CVPR*, 2016, pp. 1874–1883.

[26] Microsoft. (2023) ICASSP Acoustic Echo Cancellation Challenge. [Online]. Available: https://www.microsoft.com/en-us/research/academic-program/acoustic-echo-cancellation-challenge-icassp-2023/

[27] Microsoft. (2023) ICASSP 5th Deep Noise Suppression Challenge. [Online]. Available: https://www.microsoft.com/en-us/research/academic-program/deep-noise-suppression-challenge-icassp-2023/

[28] J. G. Beerends, N. M. P. Neumann, E. L. van den Broek, A. Llagostera Casanovas, J. T. Menendez, C. Schmidmer, and J. Berger, "Subjective and Objective Assessment of Full Bandwidth Speech Quality," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 440–449, 2020, conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[29] S. Braun, H. Gamper, C. K. Reddy, and I. Tashev, "Towards Efficient Models for Real-Time Deep Noise Suppression," in *Proceedings of ICASSP*. IEEE, 2021, pp. 656–660.

[30] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proceedings of ICLR*, 2016.

[31] M. Purin, S. Sootla, M. Sponza, A. Saabas, and R. Cutler, "AEC-MOS: A Speech Quality Assessment Metric for Echo Impairment," in *Processing of ICASSP*. IEEE, 2022, pp. 901–905.

[32] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper *et al.*, "ICASSP 2022 Deep Noise Suppression Challenge," in *Processing of ICASSP*. IEEE, 2022, pp. 9271–9275.

[33] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors," in *Proceedings of ICASSP*. IEEE, 2022, pp. 886–890.

[34] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proceedings of ICLR*, 2017.