E-URES 2.0: Efficient User-Centric Residual-Echo Suppression with a Lightweight Neural Network

Amir Ivry   Israel Cohen

Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology
Haifa 3200003, Israel
sivry@technion.ac.il   icohen@ee.technion.ac.il

Abstract—We recently introduced the Efficient User-centric Residual-Echo Suppression (E-URES) framework, which significantly reduces the floating-point operations per second (FLOPS) required during inference by 90% compared to the URES framework. The E-URES operates based on a user-operating point (UOP) defined by two key metrics: the residual echo suppression level (RESL) and the desired-speech maintained level (DSML) that the user anticipates from the output signal of a residual echo suppression (RES) system. In the first stage, an ensemble of 101 branches is employed, where each branch has two cascaded neural networks: a preliminary RES system with a design parameter, which varies between branches and balances the RESL and DSML of its RES systems' prediction, and a subsequent UOP estimator. In the second stage, a neural network uses available acoustic signals and the UOP to predict which three branches achieve the highest acoustic echo cancellation mean opinion score (AECMOS) within a specified UOP-error tolerance. Then, costly AECMOS calculations are performed only for these selected branches. Despite this efficiency mechanism, the E-URES can apply real-time inference only with dedicated and expensive hardware, limiting its wide adoption. Here, we present E-URES 2.0, which focuses on reducing the computational costs of E-URES in its first stage. A lightweight neural network preprocesses available acoustic signals and the UOP to track a subset of the 101 design parameters that their branches produce the most accurate UOP estimations in their outcomes. Only these branches are calculated during inference and continue to the AECMOS estimation stage. With 60 hours of data, we show that with a negligible performance drop on average, the E-URES 2.0 can reduce 87% of the branches and 61% of the FLOPS of the E-URES and can achieve real-time inference with standard, affordable hardware.

Index Terms—Residual-echo suppression, user-centric, AECMOS, computational efficiency, deep learning.

I. INTRODUCTION

The rise in virtual conferencing has greatly increased the utilization of hands-free voice communication [1]–[5]. In this scenario, there are typically two main points of communication: the far-end and the near-end. Generally, the distant speakers, often in a close-talk setting, have their information transmitted to the near-end speakers, who are typically in a conference room. At the near-end, the distant signal is often played through a loudspeaker placed near the local microphone [6]. During periods of simultaneous talking, the local microphone picks up the intended speech from the local participants, along with an amplified, nonlinear, echoing modification of the distant signal and background noise, which reduces the intelligibility of the conversation perceived in the far-end [7]–[9].

Most studies on RES prioritize benchmarking of different models over user preferences, as reinforced in recent research [10]–[14]. These studies often do not support a framework that balances residual echo and speech distortion, accommodates user inputs, and optimizes the AECMOS [15], which correlates strongly with subjective human ratings of speech quality in RES systems [16]. To address these gaps, we introduced the URES framework [17] and its more computationally efficient version, the E-URES [18]. Given a UOP as an input that includes the desired RESL and DSML values [16], [19] at the E-URES outcome, the E-URES starts with an ensemble of 101 branches, each containing two cascaded neural networks: an RES system with a design parameter, which varies between branches to balance the RESL and DSML of its RES systems' prediction, followed by a UOP estimator. In the second stage, a neural network uses available acoustic signals and the UOP to predict which three branches achieve the highest AECMOS and also comply with a specified UOP-error tolerance. The costly AECMOS calculations are performed only for these selected branches, and the RES system prediction with the highest AECMOS is chosen to be communicated to the far-end. The E-URES framework delivers three primary benefits: it ensures the predicted RESL and DSML are closely matched to the UOP, adjusts to any modifications in the UOP on the fly, and optimizes the AECMOS value of its output.

Although the E-URES reduces the URES' FLOPS by 90% during inference, it still requires expensive, dedicated hardware for real-time performance. It cannot achieve this on standard hardware, limiting its wider adoption. To overcome this, we introduce E-URES 2.0, which aims to reduce the number of branches calculated during inference. It utilizes a lightweight neural network that preprocesses available acoustic signals and the UOP to track a small subset of the 101 possible design parameters that their branches produce the most accurate UOP estimations at their end. Only these branches are calculated and proceed to have AECMOS estimations as in the E-URES. With the AEC-challenge database [20] and independent recordings [21], we use 60 hours to demonstrate that the E-URES 2.0 can achieve real-time inference on
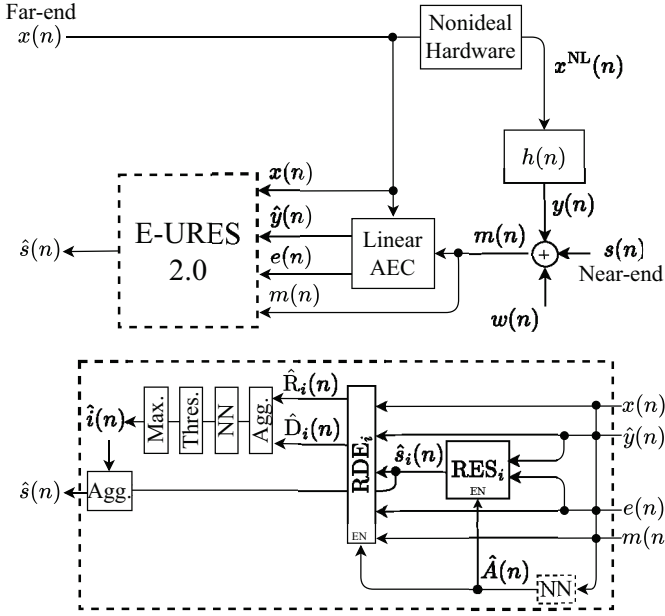
Fig. 1. The E-URES 2.0 framework. Top: a typical RES acoustic config-uration. Bottom: The E-URES 2.0 framework, featuring the neural network indicated by a dashed line. An RES and RDE pair is enabled (as indicated by the 'EN' mark) if its index is included in $\widehat{A}$. This neural network is what sets E-URES 2.0 apart from its previous version.

standard hardware by using 87% fewer branches and 61% less FLOPS, with a slight performance drop compared to the E-URES, on average. These findings encompass double-talk scenarios and hold regardless of echo-path variations, while including conditions with low signal-to-noise ratios (SNRs) and signal-to-echo ratios (SERs) that reveal our proposed framework maintains its negligible performance gap from the E-URES in challenging acoustic conditions.

II. PROBLEM FORMULATION

The E-URES 2.0 is viewed in Fig. 1. Scalars are indicated in italics, while column vectors are shown in bold. For time instance $n \in \mathbb{Z}$, the local microphone signal is:

$$m(n) = s(n) + w(n) + y(n). \tag{1}$$

Here, $s(n)$ represents the desired speech signal, $w(n)$ en-compasses environmental and system noise, and $y(n)$ holds the reverberant, nonlinearly-distorted echo from the far-end:

$$y(n) = \mathbf{h}^T(n)\,\mathbf{x}_{\mathrm{NL}}(n). \tag{2}$$

Here, $\mathbf{x}_{\mathrm{NL}}(n) \in \mathbb{R}^L$ are the recent L samples of the far-end signal after nonlinear distortions, and $\mathbf{h}(n) \in \mathbb{R}^L$ is the impulse response between the loudspeaker and micro-phone, described with L coefficients. A linear AEC is fed by $m(n)$ and the L latest samples of the remote speech signal, $\mathbf{x}(n) \in \mathbb{R}^L$, and produces an approximation of the impulse response, $\widehat{\mathbf{h}}(n) \in \mathbb{R}^L$.

The E-URES pipeline receives the echo estimate $\widehat{y}(n)$ and the linear adaptation error $e(n)$, along with $\mathbf{x}(n)$ and $m(n)$, and estimates the desired speech signal, $\widehat{s}(n)$, that is

transmitted to the far-end. The goals of the U-RES 2.0 are that $\widehat{s}(n)$ meets the UOP and an optimal AECMOS value, and to achieve real-time communication on standard hardware.

III. E-URES 2.0

A. The E-URES Framework

From this section onwards, we neglect the time index n. Consider (R, D) as the UOP, where $\mathrm{R} \in \mathbb{R}$ and $\mathrm{D} \in \mathbb{R}$ are the RESL the DSML, respectively [19]. This study supports $15 \leq \mathrm{R} \leq 30$ and $7.5 \leq \mathrm{D} \leq 15$, in dB. In [21], we introduced an RES system that operates in the STFT domain and is fed by the linear AEC's output signals [22]. A tuneable design parameter $\alpha \in \mathbb{R}$ penalizes the loss function $J(\alpha)$ during training and by that balances echo suppression and speech distortion of the RES module outcome [21]:

$$J(\alpha) = \left\|\widehat{\mathbf{S}} - \mathbf{S}\right\|_2^2 + \alpha \cdot \left\|\widehat{\mathbf{S}}\right\|_2^2 + \sigma_{\widehat{\mathbf{S}}}^2 \cdot \mathbb{I}_{\alpha>0}, \tag{3}$$

where $\widehat{\mathbf{S}}$ and $\mathbf{S}$ are respectively the amplitudes of the STFT of $\widehat{s}$ and s, $\|\cdot\|_2$ and $\sigma_{(\cdot)}^2$ are the ℓ_2 norm and variance, and $\mathbb{I}_{\alpha>0}$ equals 1 if $\alpha > 0$ and 0 otherwise. In [19], we demonstrated that the RESL tends to rise as α increases while the DSML generally declines, and the other way around. Therefore, adjusting α makes it possible to have the RESL and DSML coinciding with a given UOP. We employ 101 branches and pre-train them, where each branch starts with an RES system instance with a unique α from the set $\{0, 0.01, \ldots, 1\}$. The index i in $A = \{0, 1, \ldots, 100\}$ enumerates the RES and its prediction for every instance, denoted as RES_i and $\widehat{s}_i$, correspondingly. We use $\alpha_i = i/100$ to pre-train RES_i. Since the RESL and DSML require the desired speech signal $\mathbf{s}$ that cannot be attained during inference, each branch proceeds with an RESL-DSML Estimator (RDE) that learns to map available waveform signals to the RESL and DSML values of $\widehat{s}_i$. For each branch, RDE_i receives the waveform signals $\mathbf{x}$, $\widehat{\mathbf{y}}$, $\mathbf{e}$, $\mathbf{m}$, and $\widehat{s}_i$. The RES system RDE_i predicts the estimated RESL and DSML of $\widehat{s}_i$ as $\widehat{\mathrm{R}}_i \in \mathbb{R}$ and $\widehat{\mathrm{D}}_i \in \mathbb{R}$, respectively. We define $\Delta_{\widehat{\mathrm{R}}_i} = \left|\widehat{\mathrm{R}}_i - \mathrm{R}\right|$ and $\Delta_{\widehat{\mathrm{D}}_i} = \left|\widehat{\mathrm{D}}_i - \mathrm{D}\right|$ to capture how $\widehat{\mathrm{R}}_i \in \mathbb{R}$ and $\widehat{\mathrm{D}}_i \in \mathbb{R}$ respectively deviate from the UOP. The non-negative parameters $\mathrm{TH}_{\mathrm{R}} \in \mathbb{R}$ and $\mathrm{TH}_{\mathrm{D}} \in \mathbb{R}$, in dB, describe the upper bound allowed for these deviations. Let the subset $A^{\mathrm{TH}} \subseteq A$ hold every i value that confines to $\Delta_{\widehat{\mathrm{R}}_i} \leq \mathrm{TH}_{\mathrm{R}}$ and $\Delta_{\widehat{\mathrm{D}}_i} \leq \mathrm{TH}_{\mathrm{D}}$. Next, the waveform signals $\mathbf{x}$, $\widehat{\mathbf{y}}$, $\mathbf{e}$, $\mathbf{m}$ and $\widehat{s}_i$, and $\left(\widehat{\mathrm{R}}_i, \widehat{\mathrm{D}}_i\right)$ pairs for all i, are fed to a neural network that predicts the three branches that yield the highest AECMOS as the indices $i_p^1, i_p^2, i_p^3 \in A^{\mathrm{TH}}$. In practice, the costly AECMOS is inferred only for $\widehat{s}_{i_p^1}, \widehat{s}_{i_p^2}, \widehat{s}_{i_p^3}$. We denote $\widehat{s}_{\widehat{i}}$ as the one prediction out of the three that achieves maximal AECMOS and transmitted from the near-end.

B. The E-URES 2.0 Framework

We replace the brute-force approach of the E-URES of using 101 branches with a data-driven solution using a lean fully-connected neural network that operates in the waveform domain and has low computational demands and latency. We

first create a training set for this network by collecting from the E-URES, for every time frame, the index of the branch that produces the most accurate UOP estimate and denote it i_{UOP}, so formally $i_{\text{UOP}} = \arg\min_{i \in A} \Delta_{\widehat{R}_i} + \Delta_{\widehat{D}_i}$. Then, we train the lightweight neural network of the E-URES 2.0 to receive all possible α values, i.e., $\{0, 0.01, \ldots, 1\}$, and the waveforms $\mathbf{x}$, $\widehat{\mathbf{y}}$, $\mathbf{e}$ and $\mathbf{m}$ and map them to a vector in a one-hot-encoder representation [23], with 101 elements that all equal 0 except for the element in i_{UOP} that equals 1. During training, the categorical cross-entropy loss [24] between the ground truth and the prediction is minimized. In real-time, the network infers a vector of probabilities, and the K-top values have their branches calculated in practice, which yields K RES predictions and K corresponding UOP estimates, where $K \in A$. Formally, let $\widehat{A} = \{\widehat{i}_{\text{UOP}_1}, \widehat{i}_{\text{UOP}_2}, \ldots, \widehat{i}_{\text{UOP}_K}\}$ hold predicted indices of the K branches with the highest probability values, where $\widehat{i}_{\text{UOP}_k} \in A, \forall 1 \leq k \leq K$. Then, the calculated RES predictions and UOP estimates undergo the remainder of the pipeline similarly to the E-URES. Eventually, the RES output signal with the maximal AECMOS value within the UOP-error tolerance is delivered to the far-end.

IV. Experimental Settings

A. Data Acquisition

We utilized 50 hours of double-talk periods as both synthetic and real data from the AEC-challenge corpus [20], which features realistic clips of segments with and without changes in the echo path. No echo-path changes mean the acoustic scenario remains stationary on the local end. In contrast, periods with echo-path changes mean that the near-end setup regularly moves, whether due to devices or speakers movement, leading to concurrent filter re-convergence behavior in the linear AEC stage [8], [20], [25]. We also employed 10 hours of independent real recordings as detailed in [21], focusing on double-talk periods. This data features clips with no echo-path changes from the TIMIT [26] and Librispeech [27] databases. A mouth simulator and a loudspeaker were utilized to produce the local speech and echo, respectively, each placed at different positions in the near-end. This corpus focuses on extremely low values of SERs, extending the weakest SERs in the AEC challenge to test the operational envelope of the E-URES 2.0 performance. Overall, the SER levels, defined SER$=10\log_{10}\left[\|s(n)\|_2^2/\|y(n)\|_2^2\right]$ ranged from -20 to 10 dB, while the SNR levels, defined SNR$=10\log_{10}\left[\|s(n)\|_2^2/\|w(n)\|_2^2\right]$ ranged from 0 to 40 dB. The sample frequency for the entire database is 16 KHz.

B. Preprocessing, training, and inference

We use 45 hours from the AEC challenge for training, divided between 35 hours of real and 10 hours of synthetic recordings. From the independent recordings, we consider 5 hours for training. For testing, we take 5 hours from each corpus. We ensure that these datasets are balanced by following the practices presented in [21], e.g., for equal representation of male and female speakers, avoidance of placement of identical speakers at both ends of the conversation, and more. We partition the data into segments of 10 seconds each, which causes frequent changes in the echo path to resemble realistic cases. This leads to frequent re-convergence of the adaptive filter used to reduce linear echo, which tracks the echo-path continuously [20]. Specifically, we employ the sign-error normalized least mean square (SNLMS) adaptive filter [28]–[30] that is 150 ms in duration, i.e., $L = 2400$. Echo-paths experience abrupt changes every t seconds, with t uniformly distributed between 4 and 10 seconds, a common trait in practice. Analysis in the waveform domain is done with a 20 ms window with a 50% step size. We train the network using standard back-propagation through time and hyper-parameters that include a learning rate of 10^{-4}, which decreases by 10^{-6} every 5 epochs, 20 epochs, and a mini-batch size of 100 ms, utilizing the Adam optimizer [31]. We normalize the inputs and outputs to the network during the training phase, and apply those statistics to test data in the inference stage [32]. The neural network architecture includes an input layer, two hidden layers with 1024 neurons each, and an output layer. Each layer is followed by a dropout layer [33] with a ratio of 0.5 and a ReLU [34] activation function, except for the output layer, which uses a softmax activation [35]. The network's input layer integrates 40 ms of past context into each acoustic signal and consists of 1610 neurons. The output layer, with 101 neurons, is designed to capture the probability distribution for each of the 101 design parameters. With a 16-bit floating-point precision, the network comprises 7.8×10^6 parameters, performs 1×10^9 FLOPS, and requires for allocation and instructions an amount of 15.6 megabytes memory. Training took 18 hours on an Intel Core i7-8700K CPU @ 3.7 GHz with two Nvidia GeForce RTX 2080 Ti GPUs.

C. Performance Measures

We employ version 4 of AECMOS from Microsoft's API [15]. We report on the first category of AECMOS, which predicts how the subjective human rater would reply to, "How would you rate the echo degradation?". The AECMOS does not support shorter inputs than 15 seconds and we follow that requirement in consistency with the approach used in the URES case [17]. This metric ranges from 1 to 5 and is unitless, with 5 being the highest score. To evaluate the performance of the E-URES 2.0, we define $\Delta = \Delta_{\widehat{R}_i} + \Delta_{\widehat{D}_i}$, where a smaller Δ value indicates a more accurate UOP prediction of the output of the E-URES 2.0. To report the compute and timing requirements of the proposed framework, we consider the measures of FLOPS, memory required for allocations and instructions, number of parameters, system latency, and the real-time factor (RTF) [36].

V. Results

We report the E-URES 2.0 inference results on the entire test set using mean and standard deviation. Each waveform signal set is inferred with a random UOP from the supported values $15 \leq R \leq 30$ and $7.5 \leq D \leq 15$, in dB, to account for varied user preferences.
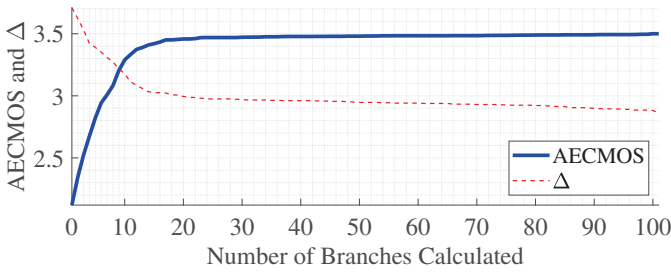
Fig. 2. The AECMOS values and the Δ of the E-URES 2.0 output prediction vs. number of branches used for inference, i.e., K, considering segments with no echo-path changes.

TABLE I
COMPUTATIONAL LOAD AND TIMING REQUIREMENTS OF THE E-URES 2.0 WITH $K = 13$ BRANCHES VS. THE E-URES.

Measure	Parameters	FLOPS	Latency	RTF
E-URES 2.0	18.1×10^6	0.56×10^{12}	15.1 ms	0.75
E-URES	26.3×10^6	1.43×10^{12}	38.6 ms	1.93

TABLE II
AECMOS RESULTS OF THE E-URES 2.0 WITH $K = 13$ BRANCHES VS. THE E-URES IN SCENARIOS WITH AND WITHOUT ECHO-PATH CHANGE.

Scenario	Echo-path change	No echo-path change
E-URES 2.0	3.02 ± 0.5	3.4 ± 0.4
E-URES	3.15 ± 0.45	3.5 ± 0.4

TABLE III
AECMOS RESULTS OF THE E-URES 2.0 WITH $K = 13$ BRANCHES VS. THE E-URES IN VARIOUS SER LEVELS WITH NO ECHO-PATH CHANGE.

SER	-20 [dB]	-10 [dB]	0 [dB]	10 [dB]
E-URES 2.0	2.85	3.31	3.70	3.91
E-URES	3.0	3.4	3.75	3.95

TABLE IV
AECMOS RESULTS OF THE E-URES 2.0 WITH $K = 13$ BRANCHES VS. THE E-URES IN VARIOUS SNR LEVELS WITH NO ECHO-PATH CHANGE.

SNR	0 [dB]	10 [dB]	25 [dB]	40 [dB]
E-URES 2.0	2.68	3.16	3.55	3.82
E-URES	2.85	3.3	3.65	3.9

In Fig 2, we examine how the AECMOS and Δ of the URES 2.0 output vary with the number of inference branches used, i.e., K. Based on these results, we chose to fix the number of inference branches to $K = 13$. Fewer branches than 13 degrade both the AECMOS and Δ to where its performance degradation does not justify its resources efficiency, while more branches are ineffective in performance but demand more redundant resources. The negligible average performance improvement of both the AECMOS and Δ from $K = 13$ to $K = 101$ demonstrates the efficiency gap the E-URES 2.0 enables compared to its previous version.

In Table I, we examine the resource demand of the E-URES 2.0 with $K = 13$ branches, and compare it with the ones of the E-URES framework. Throughout all compute and timing metrics, the E-URES 2.0 framework outperforms the E-URES in terms of efficiency, with 31% less parameters, 61% less FLOPS, and memory for allocation and instructions reduced

from over 50 megabytes to roughly 36 megabytes. When calculated using the specifications of standard and affordable hardware, i.e., the 11th Gen Intel CoreTM i7-11850H @ 2.50 GHz processor that has 0.74×10^{12} FLOPS at maximal efficiency, then accumulated buffering and inference times, i.e., latency, are cut down by 23.5 ms per input frame of 20 ms. Thus, the E-URES 2.0 achieves an RTF smaller than 1, while the E-URES has an RTF bigger than 1 and cannot operate in real-time [36]. The E-URES would require hardware with twice the amount of FLOPS than this processor to run real-time inference, while the E-URES 2.0 enables accessible and affordable adoption by personal end-users.

In Table II, we quantify how much the AECMOS and Δ values of the E-URES 2.0 were compromised on the account of its enhanced efficiency over the E-URES. Separately evaluating the AECMOS in cases with and without echo-path changes, we observe that, on average, the AECMOS decreases by 0.13 points in scenarios with echo-path changes and by 0.1 points in scenarios without echo-path changes. The standard deviation values remain nearly unchanged. Tables III and IV examine the effect of the SER and SNR levels on the E-URES 2.0 performance, only in periods without echo-path changes. Across the entire range of noise and echo levels, a negligible drop in performance is shown with SNR = 0 dB causing the maximal reduction in the AECMOS of 0.17 points, and 0.15 points when SER = −20 dB. It should be emphasized that these average gaps are mostly unperceivable to the human ear [17], and do not indicate a performance downgrade for the user at the receiving end of the conversation, i.e., the far-end. It is evident that the neural network and mechanism we introduced in E-URES 2.0 have demonstrated efficiency while maintaining nearly the same average performance as the original E-URES framework. This consistently applies to different acoustic cases, e.g., with and without echo-path changes, showcasing the neural network's generalization ability in echo-path re-convergence scenarios. Similarly, the network's robustness is indicated for varying noise and echo levels.

VI. CONCLUSIONS

The E-URES 2.0 framework has been developed as a more efficient alternative to the E-URES framework, addressing its high computational demands. A lightweight neural network associates between waveform signals and the probability of an RES system fed by those signals to minimize UOP deviations. The efficiency mechanism in this new version allows the E-URES 2.0 to reduce computational load during inference and operate in real-time on standard, readily available hardware. Our experiments indicate that E-URES 2.0 reduces computational costs by 61% with only a minor performance decrease. The E-URES 2.0 framework significantly enhances residual echo suppression, making it more practical and efficient for real-world use. This improvement boosts user experience in various acoustic environments. Moving forward, we will refine other aspects of the E-URES framework to ensure seamless integration into offices, homes, and mobile devices.

REFERENCES

[1] M. Schmidtner, C. Doering, and H. Timinger, "Agile working during COVID-19 pandemic," *IEEE Engineering Management Review*, vol. 49, no. 2, pp. 18–32, 2021.

[2] K. A. Karl, J. V. Peluchette, and N. Aghakhani, "Virtual work meetings during the COVID-19 pandemic: The good, bad, and ugly," *Small group research*, vol. 53, no. 3, pp. 343–365, 2022.

[3] A. Ratnarajah, I. Ananthabhotla, V. K. Ithapu, P. Hoffmann, D. Manocha, and P. Calamia, "Towards improved room impulse response estimation for speech recognition," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[4] Y. Hosseinkashi, L. Tankelevitch, J. Pool, R. Cutler, and C. Madan, "Meeting effectiveness and inclusiveness: large-scale measurement, identification of key features, and prediction in real-world remote meetings," *Proceedings of the ACM on Human-Computer Interaction*, vol. 8, no. CSCW1, pp. 1–39, 2024.

[5] A. Vinnikov, A. Ivry, A. Hurvitz, I. Abramovski, S. Koubi, I. Gurvich *et al.*, "Notsofar-1 challenge: New datasets, baseline, and tasks for distant meeting transcription," in *Proc. Interspeech*, 2024.

[6] K. Sridhar, R. Cutler, A. Saabas, T. Parnamaa, M. Loide, H. Gamper *et al.*, "ICASSP 2021 acoustic echo cancellation challenge: Datasets, testing framework, and results," in *Proc. ICASSP*. IEEE, 2021, pp. 151–155.

[7] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, S. L. Gay *et al.*, *Advances in network and acoustic echo cancellation*. New York: Springer, 2001.

[8] E. Hnsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. Wiley-IEEE Press, 2004.

[9] M. M. Sondhi, D. R. Morgan, and J. L. Hall, "Stereophonic acoustic echo cancellation-an overview of the fundamental problem," *IEEE Signal Processing Letters*, vol. 2, no. 8, pp. 148–151, 1995.

[10] G. Li, C. Zheng, Y. Ke, and X. Li, "Deep learning-based acoustic echo cancellation for surround sound systems," *Applied Sciences*, vol. 13, no. 3, p. 1266, 2023.

[11] T. Haubner, A. Brendel, and W. Kellermann, "End-to-end deep learning-based adaptation control for linear acoustic echo cancellation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[12] S. Xu, C. He, B. Yan, and M. Wang, "A multi-stage acoustic echo cancellation model based on adaptive filters and deep neural networks," *Electronics*, vol. 12, no. 15, p. 3258, 2023.

[13] C. Zhang, J. Liu, H. Li, and X. Zhang, "Neural multi-channel and multi-microphone acoustic echo cancellation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2181–2192, 2023.

[14] V. S. Ishwarya and M. Kothandaraman, "Novel TransQT neural network: A deep learning framework for acoustic echo cancellation in noisy double-talk scenario," *IEEE Access*, 2024.

[15] M. Purin, S. Sootla, M. Sponza, A. Saabas, and R. Cutler, "AECMOS: A speech quality assessment metric for echo impairment," in *Proc. ICASSP*. IEEE, 2022, pp. 901–905.

[16] A. Ivry, I. Cohen, and B. Berdugo, "Objective metrics to evaluate residual-echo suppression during double-talk in the stereophonic case," *Proc. Interspeech*, pp. 5348–5352, 2022.

[17] ——, "A user-centric approach for deep residual-echo suppression in double-talk," *IEEE/ACM Transations on Audio, Speech, and Language Processing*, 2024.

[18] A. Ivry and I. Cohen, "E-URES: Efficient user-centric residual-echo suppression framework with a data-driven approach to reducing computational costs," in *Proc. IWAENC*, 2024.

[19] A. Ivry, I. Cohen, and B. Berdugo, "Objective metrics to evaluate residual-echo suppression during double-talk," in *Proc. WASPAA*. IEEE, 2021, pp. 101–105.

[20] R. Cutler, A. Saabas, T. Pärnamaa, M. Purin, E. Indenbom, N. C. Ristea *et al.*, "ICASSP 2023 acoustic echo cancellation challenge," *IEEE Open Journal of Signal Processing*, pp. 1–10, 2024.

[21] A. Ivry, I. Cohen, and B. Berdugo, "Deep residual echo suppression with a tunable tradeoff between signal distortion and echo suppression," in *Proc. ICASSP*. IEEE, 2021, pp. 126–130.

[22] H. Zhivomirov, "On the development of STFT-analysis and ISTFT-synthesis routines and their practical implementation," *Technology, Education, Management, Informatics (TEM) Journal*, vol. 8, no. 1, pp. 56–64, 2019.

[23] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018.

[24] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[25] E. Seidel, P. Mowlaee, and T. Fingscheidt, "Convergence and performance analysis of classical, hybrid, and deep acoustic echo control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[26] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report*, vol. 93, p. 27403, 1993.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[28] A. Ivry, I. Cohen, and B. Berdugo, "Deep adaptation control for acoustic echo cancellation," in *Proc. ICASSP*. IEEE, 2022, pp. 741–745.

[29] N. L. Freire and S. C. Douglas, "Adaptive cancellation of geomagnetic background noise using a sign-error normalized LMS algorithm," in *Proc. ICASSP*, vol. 3. IEEE, 1993, pp. 523–526.

[30] E. Shachar, I. Cohen, and B. Berdugo, "Acoustic echo cancellation with the normalized sign-error least mean squares algorithm and deep residual echo suppression," *Algorithms*, vol. 16, no. 3, p. 137, 2023.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *preprint arXiv:1412.6980*, 2014.

[32] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *preprint arXiv:1803.08375*, 2018.

[35] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[36] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457, 2021.