SLIDING CONTEXT WINDOW POST-PROCESSING METHOD FOR NEURAL NETWORK-BASED MONAURAL SPEECH ENHANCEMENT

Luan Vinícius Fiorio[⋆]*, Boris Karanov*[⋆]*, Bo Yang*[⋆]*, Bruno Defraene*[†]*,*
Johan David[†]*, Frans Widdershoven*[†]*, Wim van Houtum*[†]*, Ronald M. Aarts*[⋆]

[⋆] Eindhoven University of Technology, Eindhoven 5600 MB, The Netherlands
[†] NXP Semiconductors, High Tech Campus 46, Eindhoven 5656 AE, The Netherlands

ABSTRACT

We apply a sliding context window post-processing to the output of a neural network-based speech enhancement (SE) model for capturing signal context dependencies between neighboring frames. In particular, the considered SE system utilizes soft masking and computes loss on the time-frequency representation of the reconstructed speech. We show that the application of a windowing function at both input and the output improves the soft mask estimation process by combining multiple estimates in different contexts. More specifically, our results show that the method increases both intelligibility and signal quality of the denoised speech in comparison with the equivalent system without sliding context window post-processing.

Index Terms— Speech enhancement, noise reduction, post-processing, neural networks, deep learning

1. INTRODUCTION

Monaural speech enhancement (SE) often concentrates on the removal of unwanted background noise from a single-channel noisy speech audio [1]. Low-complexity SE solutions might employ an acoustic environment classification scheme [2, 3] and use a noise reduction technique tailored to the specific noise type. On the other hand, considering various noise types, deep learning (DL) has been successfully applied for carrying out the SE task [4, 5, 6, 7]. Typically, DL-based speech enhancement is implemented via supervised learning, which relies on targets such as ideal binary or ideal ratio masks [4, 8]. The application of DL can lead to improved performance in terms of speech intelligibility and quality metrics when compared to classical signal processing techniques for noise reduction, such as Wiener filtering [1].

Supervised DL-based speech enhancement can operate either in time domain or time-frequency (T-F) domain. While time domain processing has attracted attention in recent years [9, 10], traditionally T-F features have been used for SE due to well established hardware implementations of the fast Fourier transform (FFT) and its inverse operation [11]. In particular, enhancement of the magnitude of T-F features is the lowest complexity processing, which however results in a suboptimal solution as it does not enhance the noisy phase.

It has been shown in the literature that additional (post) processing of the neural network (NN) outputs can further enhance the performance of the noise reduction system [12, 13, 14, 15]. In [15], for example, a post-processing method that combines gains estimated both by a NN and by a log-spectral minimum mean squared error estimator is proposed. Furthermore, in [13], a post-processing method to combine log-power spectra outputs with the ideal ratio mask outputs of a multi-objective neural network SE model is used.

In this paper, we apply a novel post-processing technique for neural network-based speech enhancement. The method aims at improving the estimated soft mask output of the neural network model. In particular, we recombine the estimated soft mask values in multiple time-dependent contexts via a sliding window post-processing. A similar method has shown promising performance for efficient signal estimation in molecular and optical communication systems [16, 17]. Our results show that, for various noise types and signal-to-noise ratios (SNR), the sliding window post-processing consistently achieves improvements both in the short-time objective intelligibility (STOI) and perceptual evaluation of speech quality (PESQ) metrics.

2. PRELIMINARIES

Let $S(t,f)$, $N(t,f)$, and $Y(t,f)$ be the short-term Fourier transform (STFT) representations of the clean speech $s(t)$, noise $n(t)$, and noisy speech signal $y(t)$, where t is the discrete time index, f is the discrete frequency index, and $Y(t,f) = S(t,f) + N(t,f)$. The ideal ratio mask (IRM) is then defined as [4]

$$M(t,f) = \left(\frac{|S(t,f)|^2}{|S(t,f)|^2 + |N(t,f)|^2} \right)^{\beta}, \qquad (1)$$

with a compression parameter $\beta \in (0,1]$. The denoised speech STFT can be obtained by

$$\hat{S}(t,f) = M(t,f) \cdot Y(t,f), \qquad (2)$$

where $(\cdot)$ represents element-wise multiplication. Notice that (2) multiplies a real-valued matrix – $M(t, f)$ – with a complex-valued matrix – $Y(t, f)$. Thus, only the magnitude of $Y(t, f)$ is denoised, while the phase is kept the same.

3. SLIDING WINDOW POST-PROCESSING

It has been shown that a sliding window technique combined with simple recurrent neural networks leads to close-to-optimal symbol sequence detection in communication systems [16, 17]. Such a scheme also allows to process the data more efficiently on a window-by-window basis, as opposed to process the whole sequence at each new data arrival.

In the framework of speech enhancement with neural networks, the sliding window technique allows to take into account a *context* of multiple neighboring STFT frames, for example in the estimation of ideal ratio masks. The output of the overall estimation process combines the soft mask estimates in different local contexts along the speech segment.

Figure 1, adapted from [17], depicts the sliding window process which we apply to the problem of speech enhancement (assuming window size $w = 3$). The next subsection describes the method in more detail.

3.1. Method description

The sliding window scheme for the neural network processor operates on the time axis of the time-frequency bin features of the noisy speech. It estimates the IRM for w bins and then slides by one bin. The resulting multi-context estimates of a mask at time t are combined to obtain a refined final mask.

Given a time duration T, the magnitude of the normalized – by it's mean and standard deviation – noisy speech STFT is denoted by the matrix $\bar{Y} = [\bar{\boldsymbol{Y}}_1 \; \bar{\boldsymbol{Y}}_2 \; ... \; \bar{\boldsymbol{Y}}_T]$, where $\bar{\boldsymbol{Y}}_t$ are vectors of F frequency features of the noisy signal at time bin t. The NN model has as inputs a context window of w time bins $[\bar{\boldsymbol{Y}}_{t-w+1} \; ... \; \bar{\boldsymbol{Y}}_t]$, estimating the corresponding w time bins of the ratio mask at the output $[\hat{\boldsymbol{M}}^{(t)}_{t-w+1} \; ... \; \hat{\boldsymbol{M}}^{(t)}_t]$. The position of the window is then shifted by one time bin, with new input window $[\bar{\boldsymbol{Y}}_{t-w+2} \; ... \; \bar{\boldsymbol{Y}}_{t+1}]$ and output estimates $[\hat{\boldsymbol{M}}^{(t+1)}_{t-w+2} \; ... \; \hat{\boldsymbol{M}}^{(t+1)}_{t+1}]$. Then, the first $w - 1$ time frames of the IRM will be estimated as [17]

$$\hat{\boldsymbol{M}}_t = \frac{1}{t} \sum_{k=1}^{t} \hat{\boldsymbol{M}}^{(k)}_t, \quad t = 1, \; ..., \; w - 1, \qquad (3)$$

where k is the (shift) iteration step of the IRM estimation. For $w \leq t \leq T - w$, the estimation of $\hat{\boldsymbol{M}}$ becomes

$$\hat{\boldsymbol{M}}_t = \frac{1}{w} \sum_{k=t-w+1}^{t} \hat{\boldsymbol{M}}^{(k)}_t, \quad t = w, \; ..., \; T - w. \qquad (4)$$

Finally, the last $w - 1$ time bins are estimated according to

$$\hat{\boldsymbol{M}}_t = \frac{1}{T - t + 1} \sum_{k=t}^{T} \hat{\boldsymbol{M}}^{(k)}_t, \;\; t = T - w + 1, \; ..., \; T. \qquad (5)$$

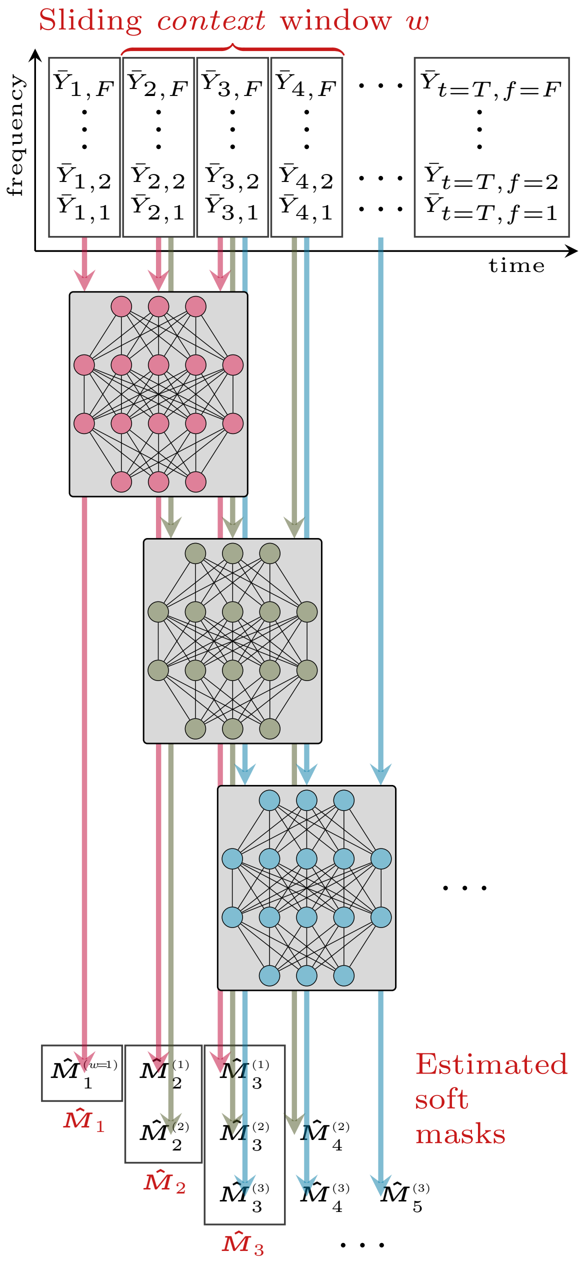


Fig. 1: Sliding context window schematic example for $w = 3$.

Notice that the sliding window method is causal, since the time context at the input of the neural network is composed of $w - 1$ past and the current bin.

3.2. Latency

A window of time bins as the input for deep neural network speech enhancement has been previously utilized, for example in [4, 5]. For real-time applications it is important to mention that this windowing increases the processing latency. The additional latency t_ℓ can be calculated as $t_\ell = (w - 1)t_{hop}$, where t_{hop} is the STFT frame hop duration in seconds. Notice that the additional latency is fixed for the whole process.

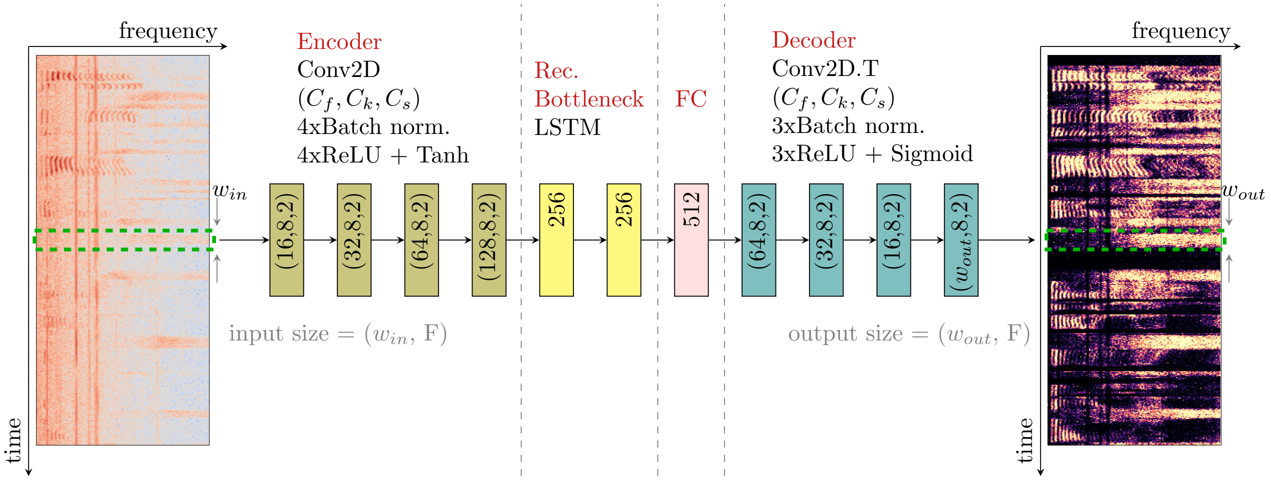


Fig. 2: Convolutional recurrent neural network schematic. C_f, C_k, and C_k are number of feature maps, kernel size, and stride.

4. SPEECH ENHANCEMENT NEURAL NETWORK

To verify our method, we adapt the convolutional recurrent network (CRN) model proposed in [5], and make it more suitable for constrained hardware implementation. The minor modifications to the model are explained in the following.

4.1. Model description

The model from [5] is a convolutional recurrent neural network composed of a convolutional encoder, recurrent bottleneck, and a convolutional decoder. In our implementation, shown in Figure 2, the convolutions are defined to only operate in the frequency dimension; the total number of layers is reduced in the encoder and decoder of the original model, also removing the skip connections; the long short-term memory (LSTM) layers have also their size reduced; and the activation functions are changed from exponential linear unit (ELU) to the simpler rectified linear unit (ReLU). The adapted model contains approximately 1.577 million parameters.

The model in Figure 2 estimates the soft masks for w_{in} consecutive frames, resulting in an output of size (w_{out}, F). For a fair comparison, we also consider a modified version of the model where the output size is $w_{out} = 1$ (no sliding window), but the input is kept $w_{in} > 1$. For the latter case, the model estimates the most recent time bin at the output.

4.2. Loss function

According to Eq. 2, the estimated ratio mask $\hat{M}(t, f)$ is applied to the noisy speech as $\hat{S}(t, f) = \hat{M}(t, f) \cdot Y(t, f)$, the result of which is used for computing the training loss. The loss function considered in training is the compressed mean square error for magnitude [18] $||\hat{S}(t, f)^{0.3} - S(t, f)^{0.3}||_2^2$, since it has been demonstrated that higher speech quality metrics can be achieved with the use of compression.

For the case where the output has only one time bin, the loss is calculated as the previously mentioned loss function, while for the sliding window case, the loss function is modified to $||\hat{S}(w, t, f)^{0.3} - S(w, t, f)^{0.3}||_2^2$, in which $\hat{S}(w, t, f)$ is a tensor with the denoised speech STFT for w frames, and $S(w, t, f)$ is the corresponding clean speech target.

5. NUMERICAL EXPERIMENTS

The experiments were carried on with the clean speech files from LibriTTS dataset [19], which is a derivation of the LibriSpeech dataset, and with the noise files from the 5th Deep Noise Suppression Challenge dataset of ICASSP 2023 [20], which is composed of various types of environment noise. The goal is to obtain a speech enhancement NN model generalized over various background noise conditions and SNRs.

5.1. Data augmentation

All audio files are initially resampled from 24 kHz to 16 kHz. Then, for the duration of a randomly drawn noise file (10 s), clean speech segments are randomly drawn and concatenated until the 10 s time is reached, where each clean speech trace receives a gain, randomly chosen, in the range of -3 to 3 dB. Fade in and fade out[1], randomly chosen in the range of 0.20 and 0.30 s, is applied to every clean speech trace, as well as for every noise file. Both 10 s noise and clean speech files are normalized by their maximum amplitude, and combined with a randomly chosen SNR within -5 and 20 dB. The noise $n(t)$, the clean speech $s(t)$, and the noisy speech $y(t)$ are then converted to the time-frequency domain with the STFT, with the following parameters: frame length of 128 samples (8 ms), frame hop of 64 samples (4 ms), and a Hanning window function. For training, 100 hours of audio from the LibriTTS ’train-clean-100’ subset are considered, while approximately 5.6 hours (1944 files) are separated for testing from the ’test-clean’ subset. The noise files are randomly (without repetition) drawn from the noise set of the DNS dataset.

[1] The fade in/out curve is obtained as $0.001e^{6.908 \cdot t}$, which allows for a dynamic range of 60 dB.

Table 1: STOI and PESQ metrics averaged over the testing set for the noisy signal and the denoised signal by the NN at -5, 0, 10, and 20 dB SNR for different window sizes. The cases where $w_{in} = w_{out}$ represent the sliding window scenario.

	-5 dB SNR		0 dB SNR		10 dB SNR		20 dB SNR	
Context window	**STOI**	**PESQ**	**STOI**	**PESQ**	**STOI**	**PESQ**	**STOI**	**PESQ**
noisy signal	0.495	1.21	0.610	1.31	0.803	1.91	0.921	3.00
$w_{in}=3$, $w_{out}=1$	0.760	1.96	0.848	2.35	0.945	3.21	0.982	3.86
$w_{in}=3$, $w_{out}=3$	0.765	1.99	0.852	2.38	0.946	3.25	0.982	3.89
$w_{in}=8$, $w_{out}=1$	0.758	1.95	0.847	2.34	0.944	3.21	0.982	3.86
$w_{in}=8$, $w_{out}=8$	0.775	2.04	0.858	2.44	0.947	3.28	0.982	3.91
$w_{in}=13$, $w_{out}=1$	0.762	1.97	0.849	2.35	0.944	3.20	0.982	3.85
$w_{in}=13$, $w_{out}=13$	0.771	1.98	0.854	2.37	0.950	3.22	0.982	3.88

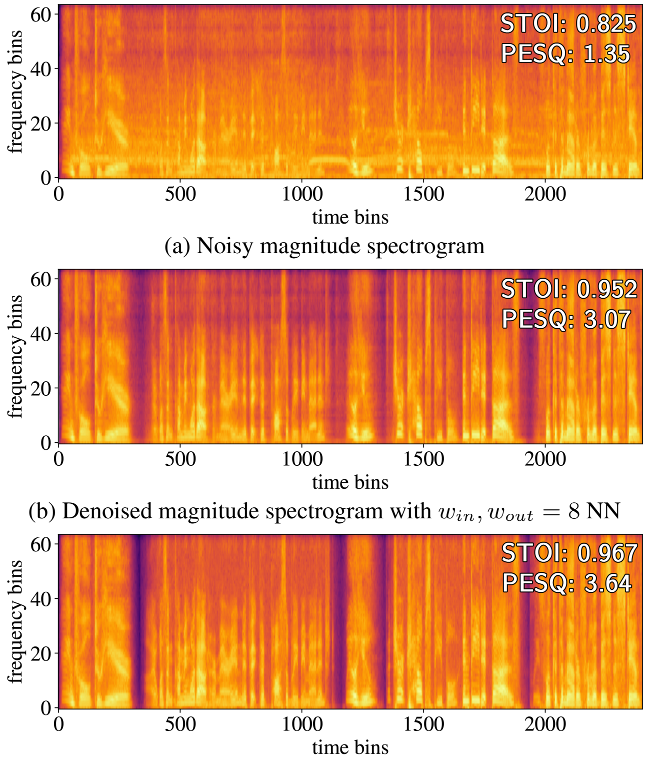
5.2. Results

We trained the model proposed in Section 4 for input windows of size $w_{in} = 3, 8, 13$, and two cases for the output: 1) $w_{out} = w_{in}$; and 2) $w_{out} = 1$. The training is executed for 50 epochs and the batch size is set to 32.

The trained models are evaluated in terms of the STOI and the PESQ metrics, with the test dataset as described in Sec. 5.1. Table 1 shows the obtained performance metrics for the noisy signal, as well as the NN-denoised audio with one bin output ($w_{out} = 1$) and the denoised audio by the NN model and sliding window post-processing ($w_{out} = w_{in}$). Note that the values shown in the table are the average over the entire test dataset.

It can be seen that the use of the sliding window improves the performance of the neural network model leading to additional gains in both STOI and PESQ in all tested SNR scenarios. The gains are larger at lower SNR, while they also depend on the size of the window. Our numerical investigation showed that a window of $w_{in}, w_{out} = 8$ frames (28 ms) achieves best results in that configuration. In particular, the largest increase in STOI is achieved at -5 dB SNR (from 0.758 to 0.775), while the largest increase in PESQ is at 0 dB SNR (from 2.34 to 2.44). Further increasing the window, does not necessarily improve the performance. On the other hand, the case of $w_{in}, w_{out} = 13$ results in a significant delay which might be prohibitive in some applications for real-time speech enhancement, such as hearing aids. For this reason, we also investigated the case of $w_{in}, w_{out} = 3$ (8 ms), which could be considered acceptable for latency-sensitive applications. As our results show, even for $w_{in}, w_{out} = 3$, the utilization of the sliding window achieves an improvement in both STOI and PESQ over the model where $w_{in} = 3, w_{out} = 1$.

Figure 3 shows an example of a single noisy file and the magnitude spectrogram of (a) the noisy audio at 10 dB, (b) the denoised audio with $w_{in}, w_{out} = 8$, and (c) the audio obtained after applying an IRM (oracle) to the noisy speech – the best possible denoising achievable without enhancing the phase. The speech in the noisy audio is composed by three different female voices and the background noise is the sound of a crowd. It is clear that the network with sliding context window post-processing is able to obtain a satisfactory noise suppression and speech preservation.



(a) Noisy magnitude spectrogram

(b) Denoised magnitude spectrogram with $w_{in}, w_{out} = 8$ NN

(c) IRM-obtained magnitude spectrogram

Fig. 3: Example of magnitude spectrograms for the noisy audio at 10 dB SNR, its denoised version by the NN with sliding window, and the IRM-obtained magnitude spectrogram.

6. CONCLUSION

We investigated the improvement in speech enhancement performance achieved by the application of a simple sliding context window post-processing at the output of the neural network. The results shows that the technique can be beneficial at a wide range of SNRs, with highest gains achieved at the most challenging (noisy) cases of -5 and 0 dB. The highest achieved gain was for $w_{in}, w_{out} = 8$ at -5 dB SNR, from 0.758 to 0.775 STOI and from 1.95 to 2.04 PESQ. Moreover, even the short window – low latency implementation (8 ms) – achieves consistently improved STOI and PESQ. For a future work, the implementation of learnable weights for the sliding window combination could be considered.

7. REFERENCES

[1] Philipos C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Inc., USA, 2nd edition, 2013.

[2] Anusha Yellamsetty, Erol J. Ozmeral, Robert A. Budinsky, and David A. Eddins, "A comparison of environment classification among premium hearing instruments," *Trends in Hearing*, vol. 25, pp. 2331216520980968, 2021, PMID: 33749410.

[3] Luan Vinícius Fiorio, Boris Karanov, Johan David, Wim van Houtum, Frans Widdershoven, and Ronald M. Aarts, "Semi-supervised learning with per-class adaptive confidence scores for acoustic environment classification with imbalanced data," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[4] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[5] Ke Tan and DeLiang Wang, "A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement," in *Proc. Interspeech 2018*, 2018, pp. 3229–3233.

[6] Rizwan Ullah, Lunchakorn Wuttisittikulkij, Sushank Chaudhary, Amir Parnianifard, Shashi Shah, Muhammad Ibrar, and Fazal-E Wahab, "End-to-end deep convolutional recurrent models for noise robust waveform speech enhancement," *Sensors*, vol. 22, no. 20, 2022.

[7] Peter Ochieng, "Deep neural network techniques for monaural speech enhancement: state of the art analysis," *arXiv preprint 2212.00369*, 2023.

[8] DeLiang Wang, *On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis*, pp. 181–197, 01 2006.

[9] Ashutosh Pandey and DeLiang Wang, "Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6875–6879.

[10] Zhifeng Kong, Wei Ping, Ambrish Dantrey, and Bryan Catanzaro, "Speech denoising in the waveform domain with self-attention," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7867–7871.

[11] Jacob Benesty, Jingdong Chen, and Emanul A.P. Habets, *Speech Enhancement in the STFT Domain*, Springer Publishing Company, Incorporated, 1st edition, 2011.

[12] Zhang Huimin, Jia Xupeng, and Li Dongmei, "An iterative post-processing approach for speech enhancement," in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, New York, NY, USA, 2019, ICMSSP '19, p. 130–134, Association for Computing Machinery.

[13] Ruwei Li, Xiaoyue Sun, Tao Li, and Fengnian Zhao, "A multi-objective learning speech enhancement algorithm based on irm post-processing with joint estimation of scnn and tcnn," *Digital Signal Processing*, vol. 101, pp. 102731, 2020.

[14] Bohan Chen, He Wang, Yue Wei, and Richard H.Y. So, "Truth-to-estimate ratio mask: A post-processing method for speech enhancement direct at low signal-to-noise ratios," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7509–7513.

[15] Feng Bao, Yuepeng Li, and Shidong Shang, "Low-complexity post-processing method for speech enhancement," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.

[16] Nariman Farsad and Andrea Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, 2018.

[17] Boris Karanov, Domaniç Lavery, Polina Bayvel, and Laurent Schmalen, "End-to-end optimized transmission over dispersive intensity-modulated channels using bidirectional recurrent neural networks," *Opt. Express*, vol. 27, no. 14, pp. 19650–19663, Jul 2019.

[18] Sebastian Braun and Ivan Tashev, "A consolidated view of loss functions for supervised deep learning-based speech enhancement," in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, 2021, pp. 72–76.

[19] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint 1904.02882*, 2019.

[20] Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, Babak Naderi, Sebastian Braun, Ross Cutler, Hannes Gamper, Mehrsa Golestaneh, and Robert Aichner, "Deep speech enhancement challenge at icassp 2023," in *ICASSP*, 2023.