Improving Robustness of Deep Learning Based Monaural Speech Enhancement Against Processing Artifacts

Ke Tan and DeLiang Wang
The Ohio State University, USA

OUTLINE

1. Background and Motivations

2. Algorithm Description

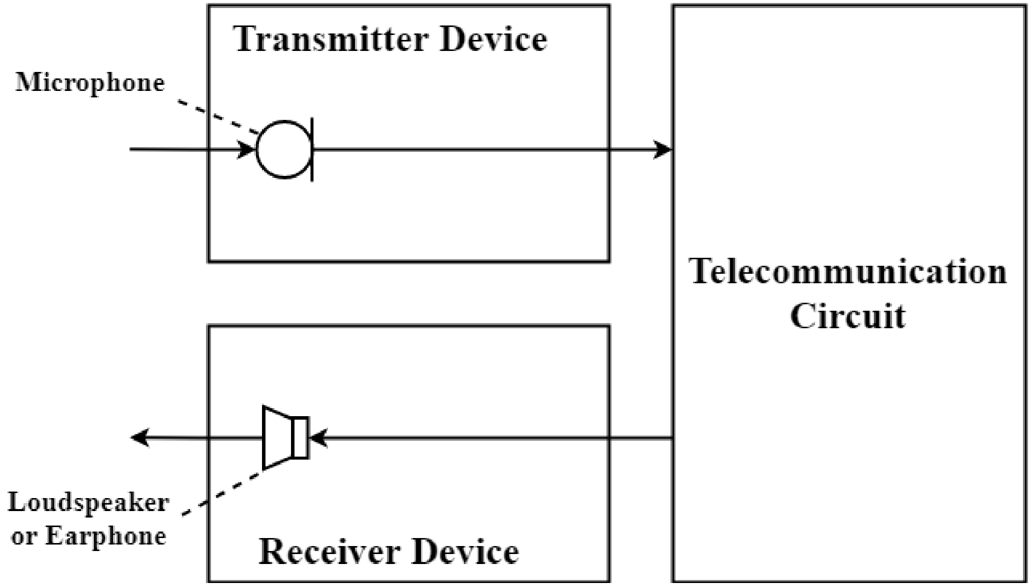
3. Evaluation and Analysis

4. Conclusion

OUTLINE

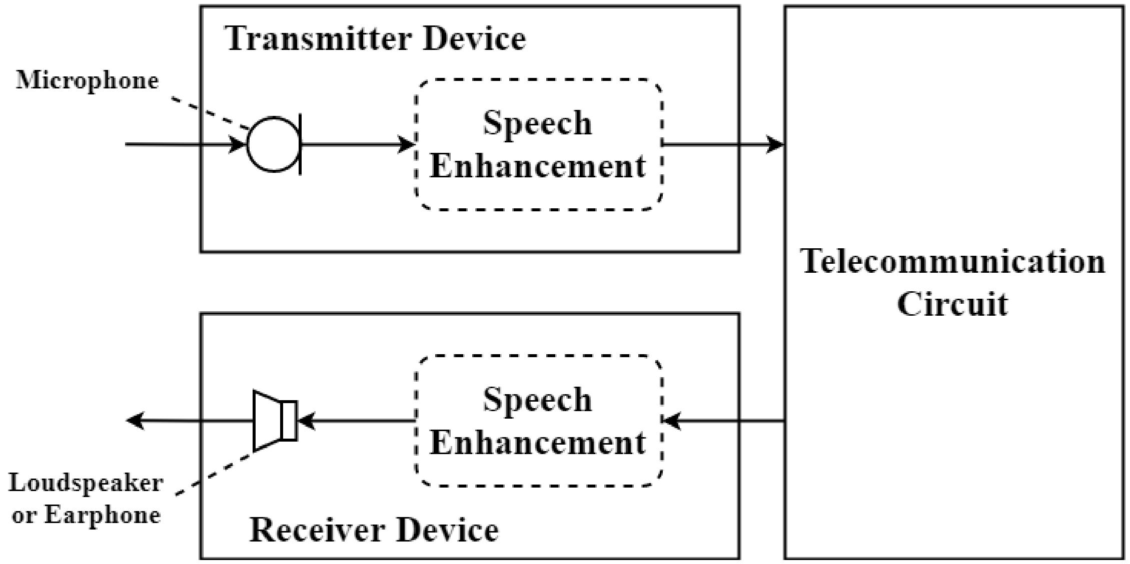
Background

- A typical voice telecommunication system consists of:
  - A transmitter (i.e. a microphone)
  - A telecommunication circuit (i.e. the physical medium that encodes and carries the speech signal)
  - A receiver (e.g. a mobile phone loudspeaker)



Background

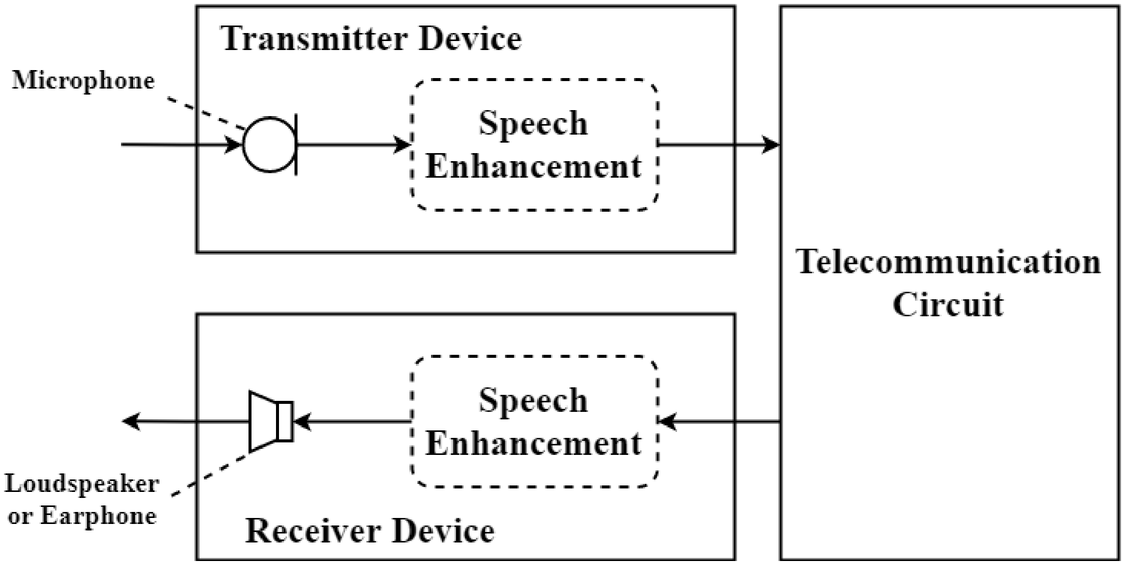
- In order to attenuate background noise, speech enhancement algorithms have been deployed in telecommunication devices.
- The speech enhancement system can be deployed in the transmitter device, the receiver device, or both.



Background

- The receiver device typically does not have the knowledge of whether speech enhancement has been performed in the transmitter device.

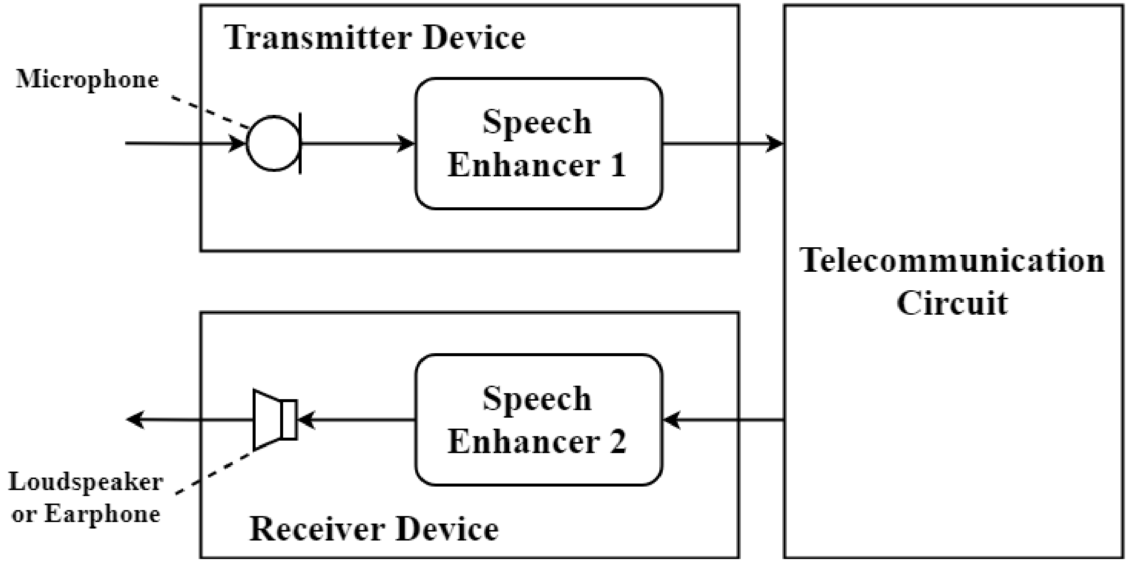
- Similarly, the transmitter device does not have the knowledge of whether the receiver device is equipped with speech enhancement.



Background

- The receiver device may choose to apply a speech enhancer to the received speech signal to cover the situation that the transmitter side lacks enhancement or its enhancement is inadequate.

- In this study, we find that enhancing noisy speech twice can be detrimental to the performance of speech enhancement. This occurs because the downstream speech enhancer is susceptible to the **processing artifacts** introduced by the upstream speech enhancer.



- Speech enhancement has been recently formulated as a supervised learning task. For any supervised learning task, generalization to untrained conditions is a crucial issue.

- In voice telecommunication, does a supervised speech enhancement model generalize to the speech signals that have been already processed by another speech enhancement algorithm?

- In this study, we investigate the processing artifacts induced by monaural speech enhancement, and their effects on a succeeding speech enhancer.

- To alleviate performance degradation caused by the processing artifacts, we propose a new training strategy for deep learning based speech enhancement in voice telecommunication.

OUTLINE

Artifacts Induced by Speech Enhancement

- Given a single-microphone mixture y, the goal of monaural speech enhancement is to separate target speech s from background noise n.

- A noisy mixture can be modeled as

$$y = s + n.$$

- Taking the time-frequency (T-F) representations of both sides, we derive

$$Y = S + N.$$

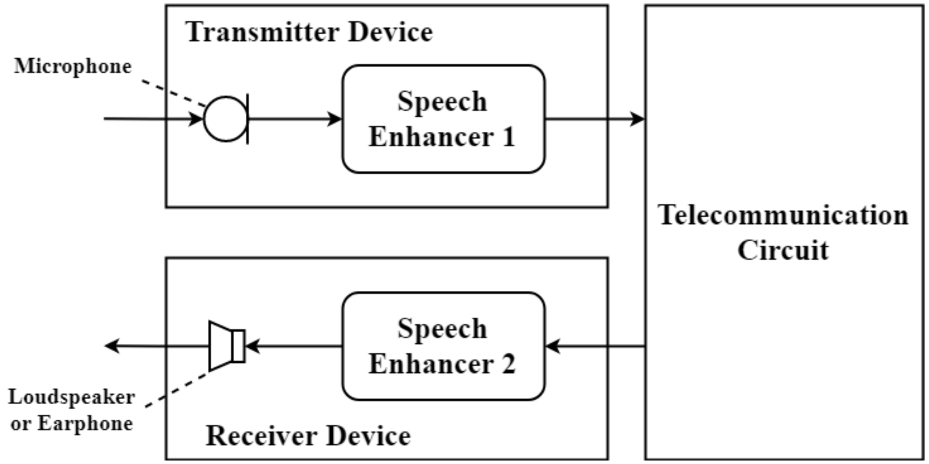
- The T-F representation $\hat{S}$ of enhanced speech can be written as:

$$\hat{S} = S + A + N^{(res)}.$$

$\hat{S}$: Enhanced Speech
S: Target Speech
A: Processing Artifact - correlated with S
$N^{(res)}$: Residual Noise - uncorrelated with S

Artifacts Induced by Speech Enhancement

- For voice telecommunication, the transmitter and receiver devices can both process a speech signal with their speech enhancers.



- If “Speech Enhancer 2” is a conventional speech enhancement method, the artifacts induced by “Speech Enhancer 1” can dissatisfy the assumptions or conditions that this enhancement method is based on.

- If “Speech Enhancer 2” is a deep learning based enhancement method, its performance can severely degrade, due to the mismatch between the pattern of enhanced speech and that of unprocessed noisy speech used for training.

Proposed Training Strategy

- To derive a robust speech enhancer against processing artifacts, we propose a new training strategy for deep learning based monaural speech enhancement.

Algorithm 1 Proposed training strategy

Input: A set of M different speech enhancers $E_j (1 \leqslant j \leqslant M)$, a randomly initialized speech enhancer E_{tr} to be trained, and a training set $T = \{(y_i, s_i)\}_{1 \leqslant i \leqslant K}$ that contains K pairs of unprocessed noisy speech y_i and clean speech s_i.

Output: A robust speech enhancer E'_{tr}.

for j in $\{1, 2, \ldots, M\}$ **do**
  for i in $\{1, 2, \ldots, K\}$ **do**
    Process y_i with E_j to produce enhanced speech $y_i^{(j)}$;
    Make a new pair of signals $(y_i^{(j)}, s_i)$;
  end for
  Collect $(y_i^{(j)}, s_i)$ for all i's into a new training set $T^{(j)} = \{(y_i^{(j)}, s_i)\}_{1 \leqslant i \leqslant K}$;
end for
Let $T' = T \cup T^{(1)} \cup T^{(2)} \cup \cdots \cup T^{(M)}$;
Train E_{tr} on the comprehensive training set T' to obtain a robust speech enhancer E'_{tr};
return E'_{tr}

Proposed Training Strategy

- We carefully choose a set of five representative traditional speech enhancement algorithms and a commonly-used feedforward DNN as E_j's:

- E_1: spectral subtraction; *- Spectral-subtractive algorithms*
- E_2: a Wiener filter based on a priori SNR estimation; *- Wiener filtering*
- E_3: an MMSE estimator;
- E_4: the IMCRA method; *- Statistical model based methods*
- E_5: a KLT-based subspace algorithm; *- Signal subspace algorithms*
- E_6: a feedforward DNN that has four hidden layers with 1024 units in each layer, where the output layer performs a spectral mapping in the magnitude domain. *- Supervised speech enhancement*

Notes:
MMSE - minimum mean-square error;
IMCRA - improved minima controlled recursive averaging;
KLT - Karhunen–Loève transform.

OUTLINE

1. Background and Motivations

2. Algorithm Description

3. Evaluation and Analysis

4. Conclusion

Experimental Setup

- Dataset: WSJ0 SI-84, including 7138 utterances from 83 speakers. Of the 83 speakers, 6 speakers (3 males and 3 females) are treated as untrained speakers for testing. The models are trained with the remaining 77 speakers.

- (1) Training noises: 10,000 noises from a sound effect library (available at https://www.sound-ideas.com). (2) Test noises: babble and cafeteria noises from an Auditec CD (available at http://www.auditec.com).

- To create a training mixture, we mix a randomly selected training utterance with a random cut from the 10,000 training noises at an SNR randomly chosen from {-8, -7, -6, -5, -4, -3, -2, -1, 0, 4, 8, 12, 16, 20} dB. We create 80,000 mixtures for training. - *“training set 1”*

- We process each mixture in *training set 1* using each of the 6 speech enhancers, i.e. spectral subtraction, Wiener filtering, MMSE, IMCRA, KLT-based subspace and a four-layer DNN. This yields a training set, which comprises 560,000 (=80,000×(1+6)) training examples. - *“training set 2”*

Experimental Setup

- We simulate a test set including 150×3 mixtures, which are created from 25×6 utterances of 6 untrained speakers. Three different SNRs are used for the test set, i.e. -5, 0 and 5 dB.

- For evaluation, we use an LSTM network with four hidden layers, as well as two newly-developed convolutional recurrent networks (CRNs) [1], [2].

- Trained on training set 1: LSTM1, CRN1 and RI-CRN1.
- Trained on training set 2: LSTM2, CRN2 and RI-CRN2.

[1] K. Tan and D. L. Wang, "A convolutional recurrent neural network for real-time speech enhancement.," in Interspeech, 2018, pp. 3229–3233.
[2] K. Tan and D. L. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6865–6869.

- Evaluations of LSTM models on the aforementioned six speech enhancers.

Table 1. Evaluation of LSTM models on different speech enhancers.

Metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
Unprocessed	57.84	69.80	81.06	1.49	1.79	2.12
LSTM1	72.82	84.98	91.57	1.88	2.39	2.80
LSTM2	**73.80**	**85.28**	**91.67**	**1.92**	**2.39**	**2.79**
Spectral subtraction [1]	56.14	70.43	82.77	1.61	1.96	2.33
Spectral subtraction - LSTM1	60.14	76.42	88.24	1.44	2.09	2.73
Spectral subtraction - LSTM2	**72.84**	**84.89**	**91.55**	**1.90**	**2.41**	**2.82**
Wiener filtering [3]	54.63	68.96	81.29	1.52	1.89	2.26
Wiener filtering - LSTM1	57.48	74.46	86.51	1.35	2.02	2.64
Wiener filtering - LSTM2	**72.50**	**84.82**	**91.57**	**1.90**	**2.40**	**2.82**
MMSE estimator [4]	54.19	67.21	79.26	1.61	1.96	2.31
MMSE estimator - LSTM1	55.55	70.27	83.27	1.41	1.96	2.57
MMSE estimator - LSTM2	**71.63**	**84.32**	**91.30**	**1.86**	**2.37**	**2.80**
IMCRA method [8]	55.33	69.50	81.56	1.54	1.90	2.27
IMCRA method - LSTM1	56.11	73.07	85.92	1.29	1.95	2.60
IMCRA method - LSTM2	**73.00**	**85.02**	**91.50**	**1.89**	**2.41**	**2.82**
KLT-based subspace [9]	55.72	71.32	83.24	1.20	1.68	2.11
KLT-based subspace - LSTM1	50.20	70.38	85.65	0.91	1.65	2.39
KLT-based subspace - LSTM2	**71.70**	**84.29**	**91.17**	**1.87**	**2.37**	**2.77**
DNN mapping	68.09	81.29	89.21	1.73	2.21	2.60
DNN mapping - LSTM1	68.78	82.37	89.76	1.69	2.26	2.69
DNN mapping - LSTM2	**71.70**	**84.29**	**91.17**	**1.87**	**2.37**	**2.77**

- STOI and PESQ evaluations on two unseen conventional speech enhancers.

Table 2. STOI and PESQ evaluations on two unseen conventional speech enhancers.

Metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
Unprocessed	57.84	69.80	81.06	1.49	1.79	2.12
LSTM1	72.82	84.98	91.57	1.88	2.39	2.80
LSTM2	**73.80**	**85.28**	**91.67**	**1.92**	**2.39**	**2.79**
CRN1 [17]	73.66	84.92	91.53	1.90	2.36	2.76
CRN2 [17]	**73.74**	**85.30**	**91.81**	**1.91**	**2.39**	**2.80**
Bayesian estimator [18]	53.16	66.45	78.56	1.58	1.95	2.33
Bayesian estimator - LSTM1	43.13	55.61	73.13	1.17	1.65	2.33
Bayesian estimator - LSTM2	**68.72**	**81.40**	**89.35**	**1.80**	**2.36**	**2.82**
Bayesian estimator - CRN1	48.81	60.68	75.14	1.05	1.44	2.08
Bayesian estimator - CRN2	**69.97**	**82.36**	**90.04**	**1.81**	**2.38**	**2.86**
Log-MMSE estimator [5]	53.75	66.98	79.09	1.52	1.89	2.26
Log-MMSE estimator - LSTM1	49.77	63.29	78.74	1.35	2.02	2.64
Log-MMSE estimator - LSTM2	**71.05**	**83.60**	**90.76**	**1.87**	**2.40**	**2.84**
Log-MMSE estimator - CRN1	53.31	65.52	79.23	1.25	1.69	2.31
Log-MMSE estimator - CRN2	**71.39**	**83.93**	**91.21**	**1.85**	**2.41**	**2.86**

- STOI and PESQ evaluations on an unseen deep learning based speech enhancer.

Table 3. STOI and PESQ evaluations on an unseen deep learning based speech enhancer.

Metrics	STOI (in %)			PESQ		
SNR	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
Unprocessed	57.84	69.80	81.06	1.49	1.79	2.12
CRN1 [17]	73.66	84.92	91.53	1.90	2.36	2.76
CRN2 [17]	**73.74**	**85.30**	**91.81**	**1.91**	**2.39**	**2.80**
RI-CRN1 [22]	76.82	87.26	93.20	2.00	2.52	2.95
RI-CRN2 [22]	**77.13**	**88.09**	**93.50**	**2.04**	**2.56**	**2.96**
LSTM masking	71.37	82.60	89.81	1.84	2.48	2.89
LSTM masking - CRN1	72.14	84.29	91.09	1.86	2.39	2.79
LSTM masking - CRN2	**72.80**	**85.13**	**91.66**	**1.86**	**2.43**	**2.85**
LSTM masking - RI-CRN1	72.88	85.67	91.97	1.84	2.48	2.89
LSTM masking - RI-CRN2	**76.72**	**87.81**	**93.14**	**2.00**	**2.58**	**2.98**

Demos

- Untrained male speaker, babble noise, -5 dB:
  - Unprocessed:
  - Wiener filtering:
  - LSTM 1:
  - LSTM 2 (Prop.):
  - Wiener filtering + LSTM 1:
  - Wiener filtering + LSTM 2:
  - Clean:

OUTLINE

1. Background and Motivations

2. Algorithm Description

3. Evaluation and Analysis

4. Conclusion

Conclusion

- In voice telecommunication, the performance of speech enhancement can severely degrade if we enhance the speech signal twice. In this study, we have examined this problem and proposed a new training strategy for the downstream speech enhancer in the receiver device.

- Our experimental results show that the proposed training strategy substantially elevate the robustness of deep learning based speech enhancement systems against processing artifacts induced by another speech enhancer.

- In addition, we find that the models trained by the proposed strategy generalize well to two new conventional speech enhancers and a new deep learning based speech enhancer.