DNN-Based Speech Presence Probability Estimation for Multi-Frame Single-Microphone Speech Enhancement

ICASSP 2020

AUD-L7: Signal Enhancement and Restoration I

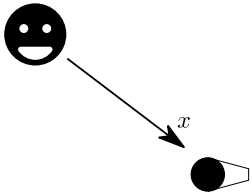
Marvin Tammen, Dörte Fischer, Bernd Meyer, Simon Doclo

Department of Medical Physics and Acoustics
Carl von Ossietzky University Oldenburg

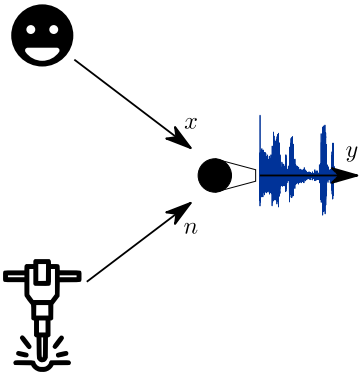
May 2020

Problem Statement

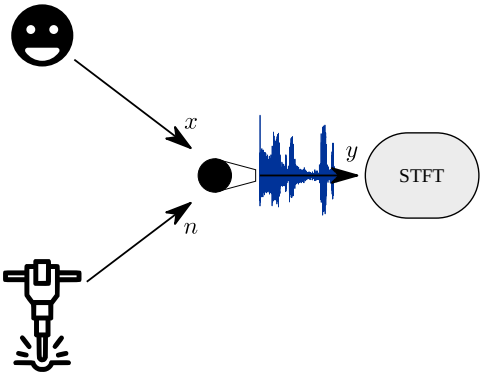
Problem Statement



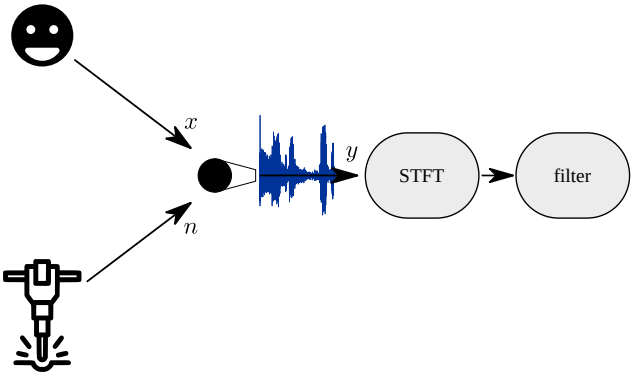
Problem Statement



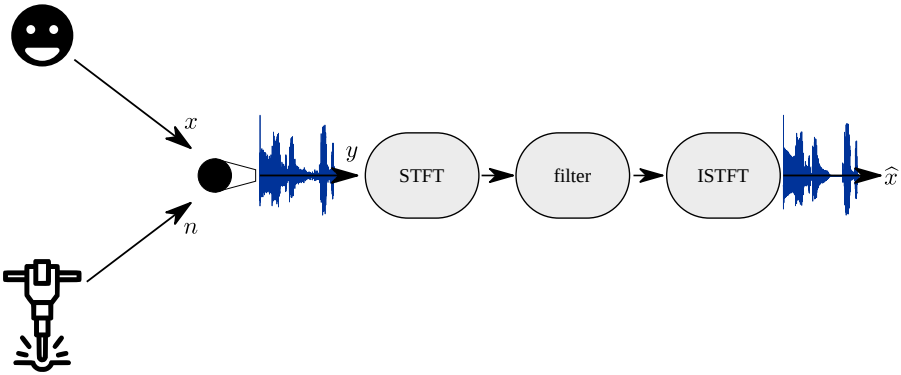
Problem Statement



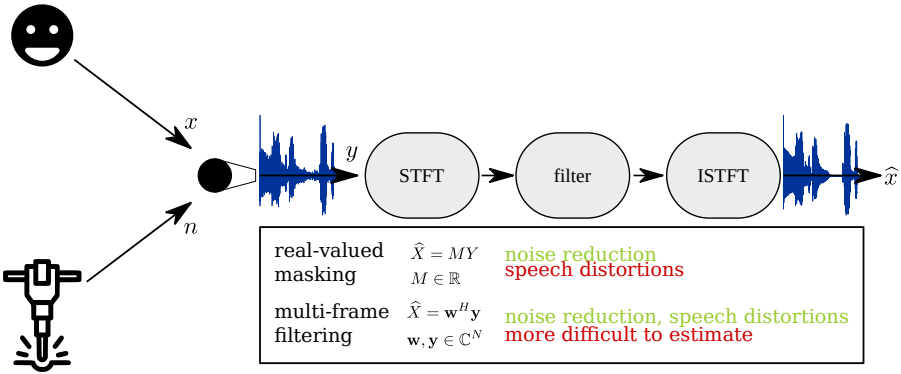
Problem Statement



Problem Statement



Problem Statement



real-valued masking	$\widehat{X} = MY$ $M \in \mathbb{R}$	noise reduction speech distortions
multi-frame filtering	$\widehat{X} = \mathbf{w}^H \mathbf{y}$ $\mathbf{w}, \mathbf{y} \in \mathbb{C}^N$	noise reduction, speech distortions more difficult to estimate

Signal Model & Assumptions

1. additive noise:
   $\mathbf{y}(k, l) = \mathbf{x}(k, l) + \mathbf{n}(k, l)$

[1] Huang and Benesty 2012.

Signal Model & Assumptions

1. additive noise:
   $\mathbf{y}(k, l) = \mathbf{x}(k, l) + \mathbf{n}(k, l)$

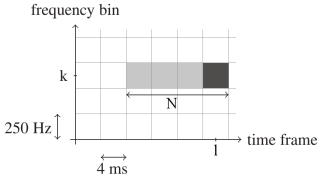


Figure: multi-frame filtering, image adapted from Fischer et al. 2016

[1] Huang and Benesty 2012.

Signal Model & Assumptions

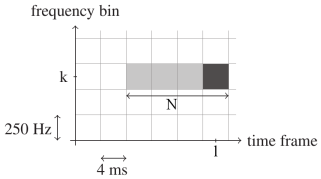


Figure: multi-frame filtering, image adapted from Fischer et al. 2016

1. additive noise:
   $\mathbf{y}(k,l) = \mathbf{x}(k,l) + \mathbf{n}(k,l)$
2. independent speech and noise:
   $\mathbf{\Phi}_{\mathbf{y}}(l) = \mathcal{E}\{\mathbf{y}(l)\mathbf{y}^H(l)\} = \mathbf{\Phi}_{\mathbf{x}}(l) + \mathbf{\Phi}_{\mathbf{n}}(l)$

[1] Huang and Benesty 2012.

Signal Model & Assumptions

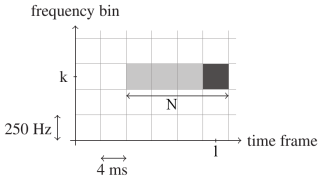


Figure: multi-frame filtering, image adapted from Fischer et al. 2016

1. additive noise:
   $\mathbf{y}(k,l) = \mathbf{x}(k,l) + \mathbf{n}(k,l)$
2. independent speech and noise:
   $\mathbf{\Phi}_{\mathbf{y}}(l) = \mathcal{E}\{\mathbf{y}(l)\mathbf{y}^H(l)\} = \mathbf{\Phi}_{\mathbf{x}}(l) + \mathbf{\Phi}_{\mathbf{n}}(l)$
3. speech has correlated and uncorrelated component[1]:
   $$\mathbf{x}(l) = \underbrace{\gamma_{\mathbf{x}}(l)X(l)}_{\text{correlated}} + \underbrace{\mathbf{x}'(l)}_{\text{uncorrelated}},$$
   $$\gamma_{\mathbf{x}}(l) = \frac{\mathcal{E}\{\mathbf{x}(l)X^*(l)\}}{\mathcal{E}\{|X(l)|^2\}} = \frac{\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}},$$
   $\gamma_{\mathbf{x}}(l)$: normalized speech interframe correlation vector

[1] Huang and Benesty 2012.

Optimization Problem and Solution

- optimization goals[2]:

[2] Huang and Benesty 2012.

Optimization Problem and Solution

- optimization goals[2]:
  1. minimize output power

$$\underset{\mathbf{w}(l)\in\mathbb{C}^N}{\operatorname{argmin}} \quad \mathbf{w}^H(l)\mathbf{\Phi}_{\mathbf{y}}(l)\mathbf{w}(l)$$

[2] Huang and Benesty 2012.

Optimization Problem and Solution

- optimization goals[2]:
  1. minimize output power
  2. while preserving speech interframe correlations (IFCs)

$$\underset{\mathbf{w}(l)\in\mathbb{C}^N}{\operatorname{argmin}} \quad \mathbf{w}^H(l)\mathbf{\Phi}_{\mathbf{y}}(l)\mathbf{w}(l), \quad \text{s.t.} \quad \mathbf{w}^H(l)\boldsymbol{\gamma}_{\mathbf{x}}(l) = 1$$

[2]Huang and Benesty 2012.

Optimization Problem and Solution

- optimization goals[2]:
  1. minimize output power
  2. while preserving speech interframe correlations (IFCs)

$$\underset{\mathbf{w}(l)\in\mathbb{C}^N}{\operatorname{argmin}} \quad \mathbf{w}^H(l)\mathbf{\Phi}_{\mathbf{y}}(l)\mathbf{w}(l), \quad \text{s.t.} \quad \mathbf{w}^H(l)\boldsymbol{\gamma}_{\mathbf{x}}(l) = 1$$

- solved by $\mathbf{w}_{\text{MFMPDR}}(l) = \dfrac{\mathbf{\Phi}_{\mathbf{y}}^{-1}(l)\boldsymbol{\gamma}_{\mathbf{x}}(l)}{\boldsymbol{\gamma}_{\mathbf{x}}^H(l)\mathbf{\Phi}_{\mathbf{y}}^{-1}(l)\boldsymbol{\gamma}_{\mathbf{x}}(l)}$

[2] Huang and Benesty 2012.

Optimization Problem and Solution

- optimization goals[2]:
  1. minimize output power
  2. while preserving speech interframe correlations (IFCs)

$$\underset{\mathbf{w}(l)\in\mathbb{C}^N}{\text{argmin}} \quad \mathbf{w}^H(l)\mathbf{\Phi}_{\mathbf{y}}(l)\mathbf{w}(l), \quad \text{s.t.} \quad \mathbf{w}^H(l)\boldsymbol{\gamma}_{\mathbf{x}}(l) = 1$$

- solved by $\mathbf{w}_{\text{MFMPDR}}(l) = \dfrac{\mathbf{\Phi}_{\mathbf{y}}^{-1}(l)\boldsymbol{\gamma}_{\mathbf{x}}(l)}{\boldsymbol{\gamma}_{\mathbf{x}}^H(l)\mathbf{\Phi}_{\mathbf{y}}^{-1}(l)\boldsymbol{\gamma}_{\mathbf{x}}(l)}$

  $\rightarrow$ estimate $\mathbf{\Phi}_{\mathbf{y}}(l), \boldsymbol{\gamma}_{\mathbf{x}}(l)$

[2] Huang and Benesty 2012.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$
  1. maximum likelihood estimator[3]

$$\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\boldsymbol{\mu}_{\mathbf{n}} \tag{1}$$

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$
  1. maximum likelihood estimator[3]

$$\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\boldsymbol{\mu}_{\mathbf{n}} \tag{1}$$

  2. noise IFC vector-based estimator

$$\widehat{\gamma}_{\mathbf{x},\gamma}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{n}}(l) \tag{2}$$

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$
  1. maximum likelihood estimator[3]

$$\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\boldsymbol{\mu}_{\mathbf{n}} \tag{1}$$

  2. noise IFC vector-based estimator

$$\widehat{\gamma}_{\mathbf{x},\gamma}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{n}}(l) \tag{2}$$

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$
  1. maximum likelihood estimator[3]

$$\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\boldsymbol{\mu}_{\mathbf{n}} \tag{1}$$

  2. noise IFC vector-based estimator

$$\widehat{\gamma}_{\mathbf{x},\gamma}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{n}}(l) \tag{2}$$

- a-priori SNR $\xi(l)$: decision-directed approach (DDA)[4]

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Speech IFC Vector

- $\gamma_{\mathbf{x}}(l) = \frac{1+\xi(l)}{\xi(l)}\gamma_{\mathbf{y}}(l) - \frac{1}{\xi(l)}\gamma_{\mathbf{n}}(l), \quad \xi(l) = \frac{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{e}}{\mathbf{e}^T\mathbf{\Phi}_{\mathbf{n}}(l)\mathbf{e}}$
  1. maximum likelihood estimator[3]

$$\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\boldsymbol{\mu}_{\mathbf{n}} \tag{1}$$

  2. noise IFC vector-based estimator

$$\widehat{\gamma}_{\mathbf{x},\gamma}(l) = \frac{1+\widehat{\xi}(l)}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{y}}(l) - \frac{1}{\widehat{\xi}(l)}\widehat{\gamma}_{\mathbf{n}}(l) \tag{2}$$

- a-priori SNR $\xi(l)$: decision-directed approach (DDA)[4]
  - requires $\mathbf{e}^T\widehat{\mathbf{\Phi}}_{\mathbf{n}}(l)\mathbf{e}$

[3] Schasse and Martin 2014.
[4] Ephraim and Malah 1984.

Correlation Matrices

- noisy: fixed recursive smoothing:

$$\widehat{\mathbf{\Phi}}_{\mathbf{y}}(l) = \lambda_y \widehat{\mathbf{\Phi}}_{\mathbf{y}}(l-1) + (1-\lambda_y)\mathbf{y}(l)\mathbf{y}^H(l) \tag{3}$$

[5] Cohen 2003.

Correlation Matrices

- noisy: fixed recursive smoothing:

$$\widehat{\mathbf{\Phi}}_{\mathbf{y}}(l) = \lambda_y \widehat{\mathbf{\Phi}}_{\mathbf{y}}(l-1) + (1-\lambda_y)\mathbf{y}(l)\mathbf{y}^H(l) \tag{3}$$

- noise: SPP-based adaptive recursive smoothing[5]:

$$\widehat{\mathbf{\Phi}}_{\mathbf{n}}(l) = \lambda_n(l)\widehat{\mathbf{\Phi}}_{\mathbf{n}}(l-1) + (1-\lambda_n(l))\mathbf{y}(l)\mathbf{y}^H(l), \quad \text{where} \tag{4}$$

$$\lambda_n(l) = \alpha_n + (1-\alpha_n)\widehat{\mathrm{SPP}}(l) \tag{5}$$

[5] Cohen 2003.

Correlation Matrices

- noisy: fixed recursive smoothing:

$$\widehat{\mathbf{\Phi}}_{\mathbf{y}}(l) = \lambda_y \widehat{\mathbf{\Phi}}_{\mathbf{y}}(l-1) + (1-\lambda_y)\mathbf{y}(l)\mathbf{y}^H(l) \tag{3}$$

- noise: SPP-based adaptive recursive smoothing[5]:

$$\widehat{\mathbf{\Phi}}_{\mathbf{n}}(l) = \lambda_n(l)\widehat{\mathbf{\Phi}}_{\mathbf{n}}(l-1) + (1-\lambda_n(l))\mathbf{y}(l)\mathbf{y}^H(l), \quad \text{where} \tag{4}$$

$$\lambda_n(l) = \alpha_n + (1-\alpha_n)\widehat{\mathrm{SPP}}(l) \tag{5}$$

$$\Rightarrow \begin{array}{ll} \text{(i): } \widehat{\mathrm{SPP}}(l) = 0 \rightarrow \lambda_n(l) = \alpha_n, & \text{update of } \widehat{\mathbf{\Phi}}_{\mathbf{n}}(l) \\ \text{(ii): } \widehat{\mathrm{SPP}}(l) = 1 \rightarrow \lambda_n(l) = 1, & \text{no update of } \widehat{\mathbf{\Phi}}_{\mathbf{n}}(l) \end{array}$$

[5] Cohen 2003.

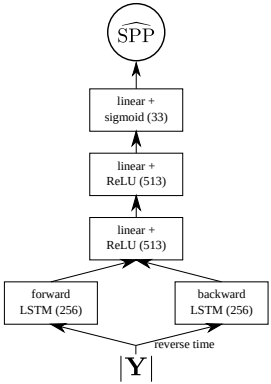
Speech Presence Probability

1. maximum likelihood-based[6]

[6] Gerkmann and Hendriks 2012.

Speech Presence Probability

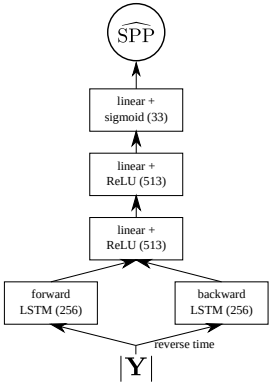
1. maximum likelihood-based[6]
2. deep recurrent neural network-based:



[6] Gerkmann and Hendriks 2012.

Speech Presence Probability

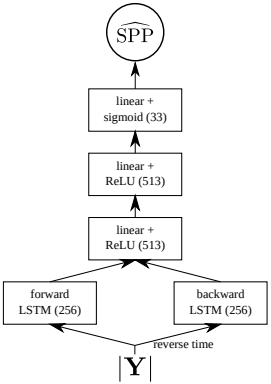
1. maximum likelihood-based[6]
2. deep recurrent neural network-based:
   - input feature: $|\mathbf{Y}| \in \mathbb{R}^{K \times L}$



[6] Gerkmann and Hendriks 2012.

Speech Presence Probability

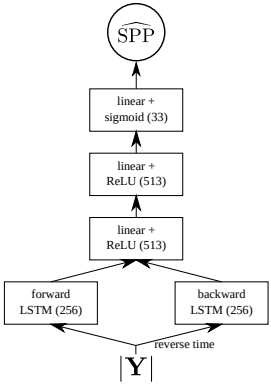
1. maximum likelihood-based[6]
2. deep recurrent neural network-based:
   - input feature: $|\mathbf{Y}| \in \mathbb{R}^{K \times L}$
   - sigmoid output to ensure $\widehat{\mathrm{SPP}}(l) \in]0, 1[$



[6] Gerkmann and Hendriks 2012.

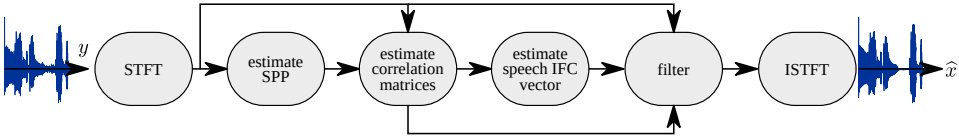
Speech Presence Probability

1. maximum likelihood-based[6]
2. deep recurrent neural network-based:
   - input feature: $|\mathbf{Y}| \in \mathbb{R}^{K \times L}$
   - sigmoid output to ensure $\widehat{\mathrm{SPP}}(l) \in]0, 1[$
   - trained to minimize MSE loss between estimated and target SPP[6]

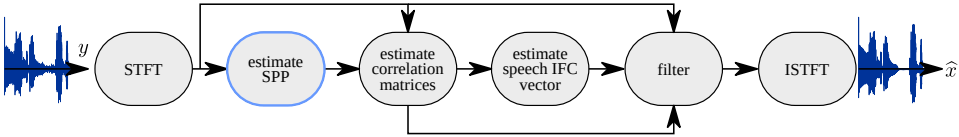


[6] Gerkmann and Hendriks 2012.

Method Overview

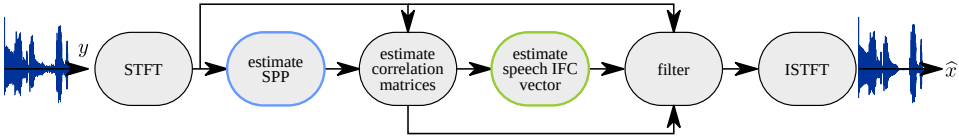


Method Overview



estimate SPP:
ML-based
vs. DNN-based

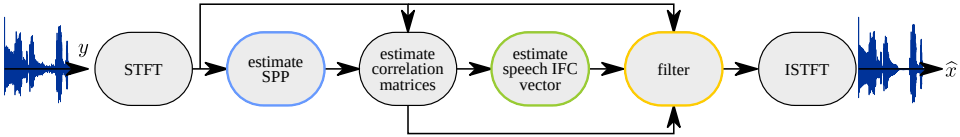
Method Overview



estimate SPP:
ML-based
vs. DNN-based

estimate speech IFC vector:
using ML constant noise IFC vector ($\boldsymbol{\mu}_{\mathbf{n}}$)
vs. using estimated noise IFC vector ($\widehat{\boldsymbol{\gamma}}_{\mathbf{n}}(l)$)

Method Overview



estimate SPP:
ML-based
vs. DNN-based

estimate speech IFC vector:
using ML constant noise IFC vector ($\boldsymbol{\mu}_{\mathbf{n}}$)
vs. using estimated noise IFC vector ($\widehat{\boldsymbol{\gamma}}_{\mathbf{n}}(l)$)

filter:
multi-frame (MPDR)
vs. single-frame Wiener gain (WG)

Datasets

	training	validation	testing
speech (WSJ0[7])	101 speakers	20 speakers	4 speakers
noise	NOISEX92[8]		Aurora[9]
SNRs / dB	[0, 20]		{-5, 0, . . . , 20}

[7] Paul and Baker 1992.
[8] Varga and Steeneken 1993.
[9] Hirsch and Pearce 2000.

Datasets

	training	validation	testing
speech (WSJ0[7])	101 speakers	20 speakers	4 speakers
noise	NOISEX92[8]		Aurora[9]
SNRs / dB	[0, 20]		{-5, 0, . . . , 20}

- disjoint training and testing datasets

[7] Paul and Baker 1992.
[8] Varga and Steeneken 1993.
[9] Hirsch and Pearce 2000.

Datasets

	training	validation	testing
speech (WSJ0[7])	101 speakers	20 speakers	4 speakers
noise	NOISEX92[8]		Aurora[9]
SNRs / dB	[0, 20]		{-5, 0, . . . , 20}

- disjoint training and testing datasets
- one DNN is trained for all SNRs

[7] Paul and Baker 1992.
[8] Varga and Steeneken 1993.
[9] Hirsch and Pearce 2000.

Settings

- STFT and algorithm settings:
  - frame length 4 ms, shift length 1 ms, Hann windows
  - recursive smoothing constants $\alpha_n \mathrel{\hat{=}} 50\,\mathrm{ms},\ \lambda_y \mathrel{\hat{=}} 12\,\mathrm{ms},\ \lambda_{\mathrm{DDA}} \mathrel{\hat{=}} 33\,\mathrm{ms}$
  - Tikhonov regularization for matrix inversion
  - filter size $N = 18$ (correlations within 21 ms can be exploited)
  - minimum gain -17 dB
- DNN training
  - dropout, batch normalization, early stopping
  - Adam optimizer
- evaluation in terms of ΔPESQ

Results

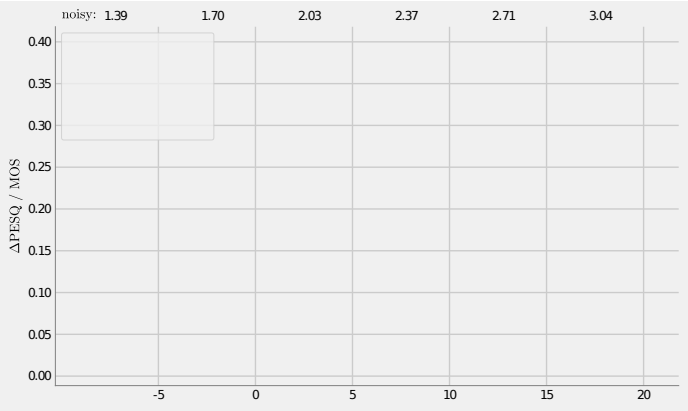


Figure: average PESQ improvement vs. input SNR / dB

Results

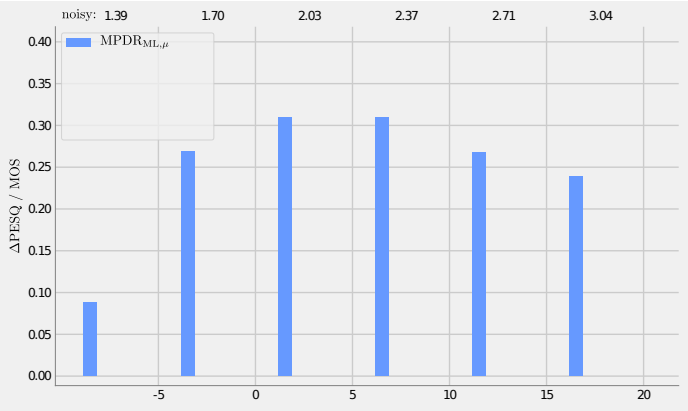


Figure: average PESQ improvement vs. input SNR / dB

Results

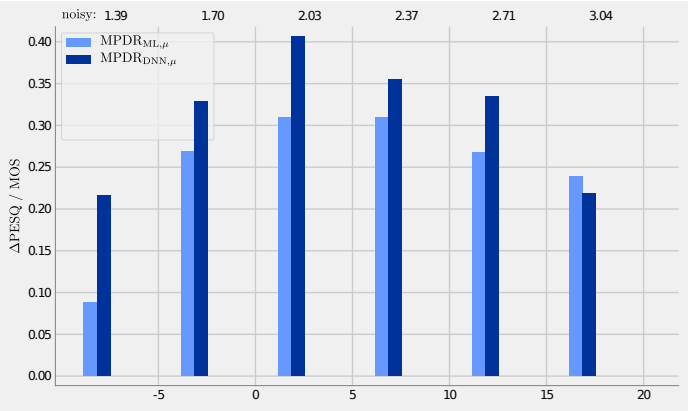


Figure: average PESQ improvement vs. input SNR / dB

Results

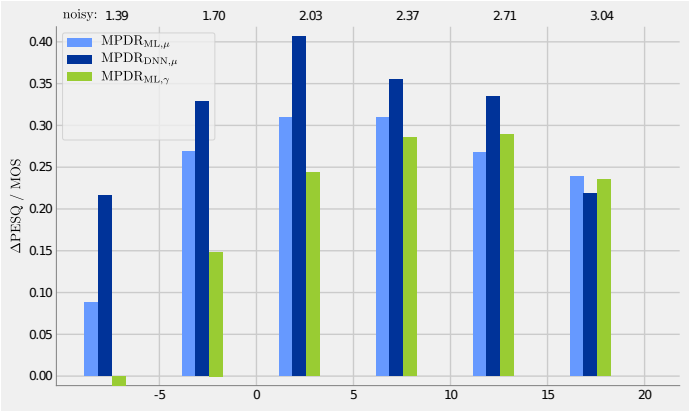


Figure: average PESQ improvement vs. input SNR / dB

Results

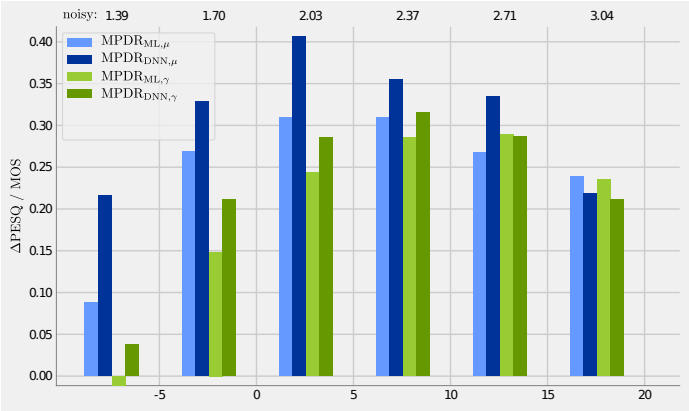


Figure: average PESQ improvement vs. input SNR / dB

Results

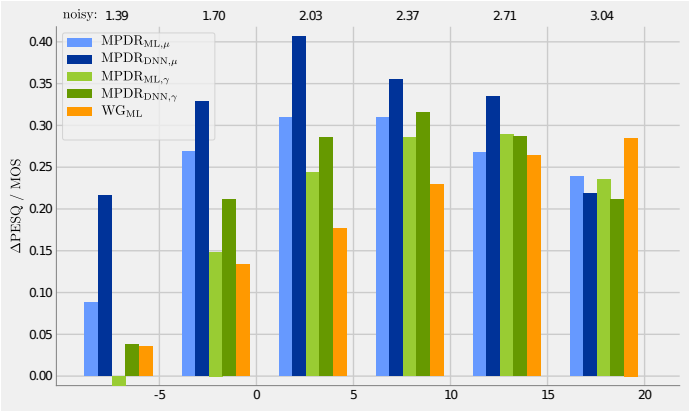


Figure: average PESQ improvement vs. input SNR / dB

Results

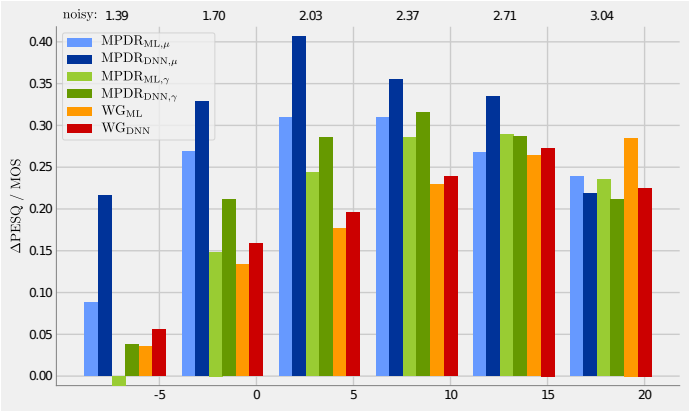


Figure: average PESQ improvement vs. input SNR / dB

Results

- $\text{MPDR}_{\text{DNN},\mu}$ with highest performance

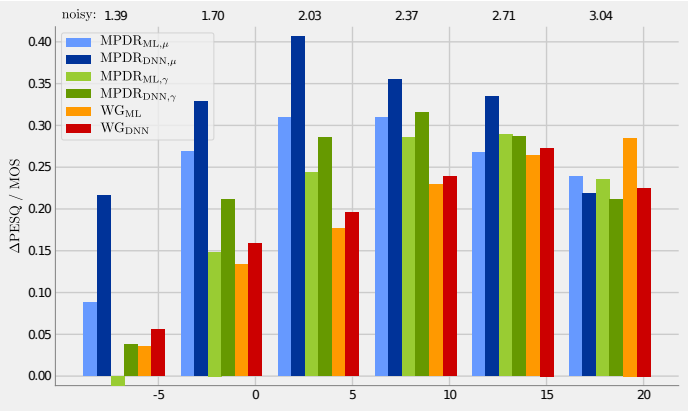


Figure: average PESQ improvement vs. input SNR / dB

Results

- $\text{MPDR}_{\text{DNN},\mu}$ with highest performance
- SPP estimation: DNN > ML

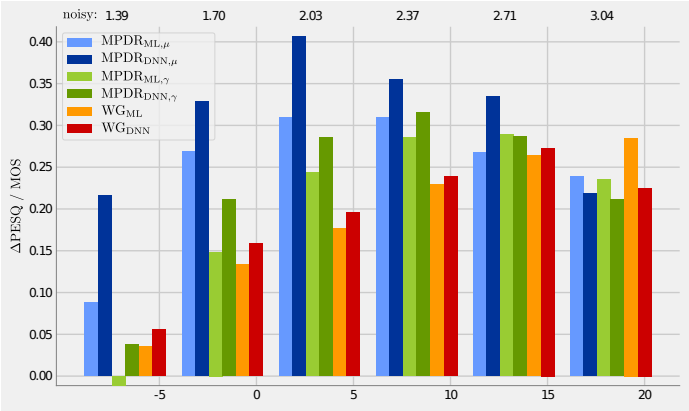


Figure: average PESQ improvement vs. input SNR / dB

Results

- $\mathrm{MPDR}_{\mathrm{DNN},\mu}$ with highest performance
- SPP estimation: DNN $>$ ML
- speech IFC estimation: using $\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) > \widehat{\gamma}_{\mathbf{x},\boldsymbol{\gamma}}(l)$

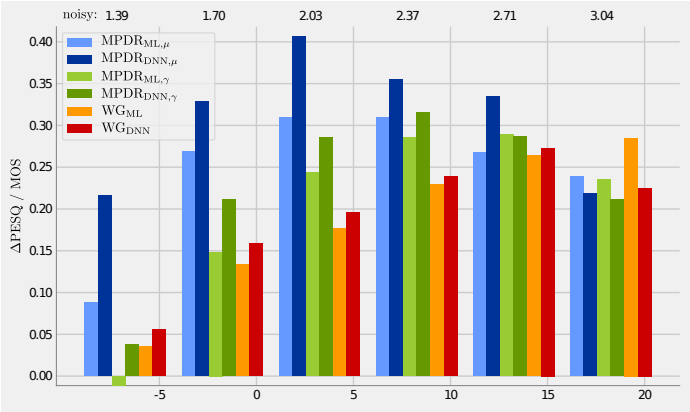


Figure: average PESQ improvement vs. input SNR / dB

Results

- $\text{MPDR}_{\text{DNN},\mu}$ with highest performance
- SPP estimation: DNN $>$ ML
- speech IFC estimation: using $\widehat{\gamma}_{\mathbf{x},\boldsymbol{\mu}}(l) > \widehat{\gamma}_{\mathbf{x},\boldsymbol{\gamma}}(l)$
- filtering: multi-frame $>$ single-frame

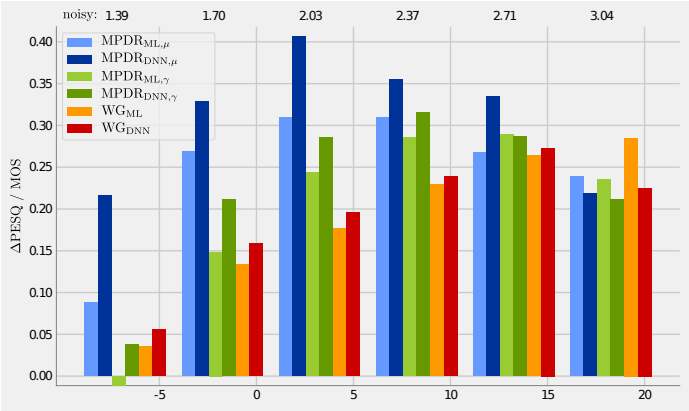


Figure: average PESQ improvement vs. input SNR / dB

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach
- **improved SE performance** by replacing conventional ML-based SPP estimator **with DNN-based SPP estimator**

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach
- **improved SE performance** by replacing conventional ML-based SPP estimator **with DNN-based SPP estimator**
- confirmed that **multi-frame filtering** can be **advantageous to single-frame filtering**

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach
- **improved SE performance** by replacing conventional ML-based SPP estimator **with DNN-based SPP estimator**
- confirmed that **multi-frame filtering** can be **advantageous to single-frame filtering**

- tight integration of deep learning with filtering

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach
- **improved SE performance** by replacing conventional ML-based SPP estimator **with DNN-based SPP estimator**
- confirmed that **multi-frame filtering** can be **advantageous to single-frame filtering**

- tight integration of deep learning with filtering
  - include filter in training process

Conclusions & Outlook

- considered **multi-frame single-microphone speech enhancement** (SE) approach
- **improved SE performance** by replacing conventional ML-based SPP estimator **with DNN-based SPP estimator**
- confirmed that **multi-frame filtering** can be **advantageous to single-frame filtering**

- tight integration of deep learning with filtering
  - include filter in training process
  - more strongly rely on DNN for parameter estimation

Thank you for your attention!

Any questions?

⇒✉marvin.tammen@uol.de

Thank you for your attention!

Any questions?

⇒✉marvin.tammen@uol.de

Maximum Likelihood-Based SPP Computation

- assumptions[10]:
  - complex Gaussian distributions for microphone, speech and noise STFT coefficients
  - equal a-priori probability of speech and noise
  - $\rightarrow$ derive likelihood under speech presence ($\mathcal{H}_1$) and absence ($\mathcal{H}_0$)
  - typical SNR under speech presence: $\xi_{\mathcal{H}_1} = 15\,\mathrm{dB}$
- resulting a-posteriori SPP estimate:

$$\widehat{\mathrm{SPP}}(l) = \left(1 + \frac{P(\mathcal{H}_0)}{P(\mathcal{H}_1)}(1+\xi_{\mathcal{H}_1})e^{-\frac{|Y(l)|^2}{\widehat{\phi}_N(l-1)}\frac{\xi_{\mathcal{H}_1}}{1+\xi_{\mathcal{H}_1}}}\right)^{-1} \tag{6}$$

[10] Gerkmann and Hendriks 2012.

Training and Testing

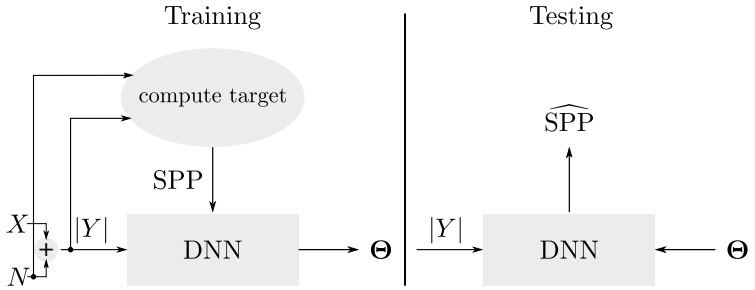


Figure: training and testing phase; Θ: DNN parameters

Wiener Gain Implementation

Wiener Gain

Input: noisy STFT coefficients $Y(k,l)$, noise PSD estimate $\widehat{\phi}_n(k,l)$, smoothing constant α, minimum gain $G_{\min}$

Output: clean speech estimate $\widehat{X}(k,l)$

foreach $k \in [0, K-1]$ **do**

  init. $\widehat{X}(k,0) = 0$;

  foreach $l \in [1, L-1]$ **do**

    $\widehat{\xi}(k,l) = \alpha \frac{|\widehat{X}(k,l-1)|^2}{\widehat{\phi}_n(k,l-1)} + (1-\alpha)\frac{|Y(k,l)|^2}{\widehat{\phi}_n(k,l)}$; `/* a-priori SNR estimation */`

    $G(k,l) = \max\left\{\frac{\widehat{\xi}(k,l)}{1+\widehat{\xi}(k,l)}, G_{\min}\right\}$; `/* spectral flooring */`

    $\widehat{X}(k,l) = G(k,l)Y(k,l)$;

  end

end

Tikhonov Regularization

- useful for dealing with multicollinearity in linear regression problems
- obtain Tikhonov-regularized matrix $\tilde{\mathbf{A}}$ from $\mathbf{A} \in \mathbb{C}^{N \times N}$ as

$$\tilde{\mathbf{A}} = \mathbf{A} + \frac{\delta}{N} \mathrm{trace}\{|\mathbf{A}|\} \mathbf{I}_N \tag{7}$$