Joint DNN-Based Multichannel Reduction of Acoustic Echo, Reverberation and Noise: Supporting Document

Guillaume Carbajal, Romain Serizel, Emmanuel Vincent, Eric Humbert

Joint DNN-Based Multichannel Reduction of Acoustic Echo, Reverberation and Noise: Supporting Document

Guillaume Carbajal, Romain Serizel , Emmanuel Vincent , Éric Humbert

RESEARCH

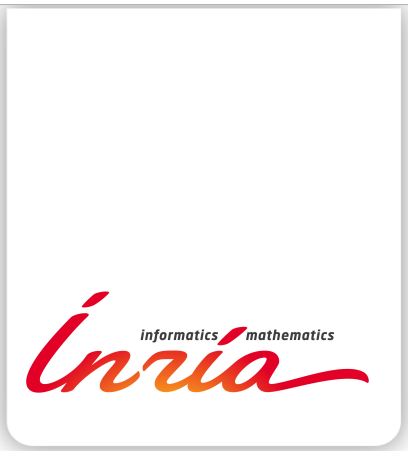
REPORT

N° 9303

November 2019

Project-Team Multispeech

ISSN 0249-6399 ISRN INRIA/RR--9303--FR+ENG



Joint DNN-Based Multichannel Reduction of Acoustic Echo, Reverberation and Noise: Supporting Document

Guillaume Carbajal *†, Romain Serizel *, Emmanuel Vincent *,
Éric Humbert †

Project-Team Multispeech

Research Report n° 9303 — November 2019 — 32 pages

Abstract: This technical report is the supporting document of our proposed approach based on deep neural networks (DNNs) for joint multichannel reduction of echo, reverberation and noise [1]. First, this report explains in detail the derivation of the update rules of the iterative block-coordinate ascent algorithm which jointly optimizes all the distortion-specific filters, and provides the detailed pseudo-code of this algorithm. Secondly, this report presents the detailed pseudo-code of the iterative algorithm which derives the targets for the neural network used in the approach. Finally, this report give details of the recording and simulation parameters of the dataset.

Key-words: Acoustic echo, reverberation, background noise, joint distortion reduction, expectation-maximization, recurrent neural network.

* Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
† Invoxia SAS, 8 esplanade de la Manufacture, 92130 Issy-les-Moulineaux, France

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique

CS20101

54603 Villers-lès-Nancy Cedex

Contents

This technical report is organized as follows. In Section I, we recall the model of the proposed approach. We express the vectorized computation of echo cancellation and dereverberation in Section II. In section III we detail the complete derivation of the update rules. Section IV describes the computation of the ground truth targets for the neural network used in our approach. Finally Section V specifies the recording and simulation parameters of the dataset, such as the positions of the speakers and the acoustic properties of the rooms.

1 Problem, model and expression of the likelihood

We adopt the following general notations: scalars are represented by plain letters, vectors by bold lowercase letters, and matrices by bold uppercase letters. The symbol $(\cdot)^*$ refers to complex conjugation, $(\cdot)^T$ to matrix transposition, $(\cdot)^H$ to Hermitian transposition, $\mathrm{tr}(\cdot)$ to the trace of a matrix, $\|\cdot\|$ to the Euclidean norm and $\otimes$ to the Kronecker product. The identity matrix is denoted $\mathbf{I}$. Its dimension is either implied by the context or explicitly specified by a subscript.

1.1 Problem

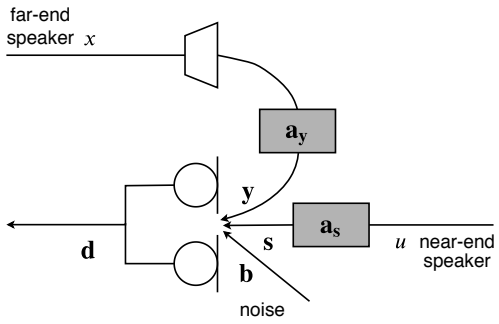


Figure 1: Acoustic echo, reverberation and noise problem.

In real scenarios, acoustic echo, near-end reverberation and background noise can be simultaneously present as illustrated in Fig. 1. Here, the conditions are assumed to be time-invariant, i.e., the positions of the recording device and the speaker and the room acoustics do not change over the duration of an utterance. Denoting by M the number of channels (microphones), the mixture $\mathbf{d}(t) \in \mathbb{R}^{M\times 1}$ observed at the microphones at time t is the sum of the near-end signal

$\mathbf{s}(t) \in \mathbb{R}^{M\times 1}$, the acoustic echo $\mathbf{y}(t) \in \mathbb{R}^{M\times 1}$, and a noise signal $\mathbf{b}(t) \in \mathbb{R}^{M\times 1}$:

$$\mathbf{d}(t) = \mathbf{s}(t) + \mathbf{y}(t) + \mathbf{b}(t). \tag{1}$$

The acoustic echo $\mathbf{y}(t)$ is a nonlinearly distorted version of the observed far-end signal $x(t) \in \mathbb{R}$ played by the loudspeaker, which is assumed to be single-channel. The linear part can be approximated by the linear convolution of $x(t)$ and the M-dimensional room impulse response (RIR) $\mathbf{a}_{\mathrm{y}}(\tau) \in \mathbb{R}^{M\times 1}$, or echo path, modeling the acoustic path from the loudspeaker (including the loudspeaker response) to the microphones. The echo signal can be expressed as

$$\mathbf{y}(t) \approx \sum_{\tau=0}^{\infty} \mathbf{a}_{\mathrm{y}}(\tau) x(t-\tau), \tag{2}$$

The reverberant near-end signal $\mathbf{s}(t)$ is obtained by linear convolution of the anechoic near-end signal $u(t) \in \mathbb{R}$ and the M-dimensional RIR $\mathbf{a}_{\mathrm{s}}(\tau) \in \mathbb{R}^{M\times 1}$

$$\mathbf{s}(t) = \sum_{\tau=0}^{\infty} \mathbf{a}_{\mathrm{s}}(\tau) u(t-\tau). \tag{3}$$

This signal can be decomposed as

$$\mathbf{s}(t) = \underbrace{\sum_{0\leq\tau\leq t_{\mathrm{e}}} \mathbf{a}_{\mathrm{s}}(\tau) u(t-\tau)}_{=\mathbf{s}_{\mathrm{e}}(t)} + \underbrace{\sum_{\tau>t_{\mathrm{e}}} \mathbf{a}_{s}(\tau) u(t-\tau)}_{=\mathbf{s}_{\mathrm{l}}(t)}, \tag{4}$$

where $\mathbf{s}_{\mathrm{e}}(t)$ denotes the early near-end signal component, $\mathbf{s}_{\mathrm{l}}(t)$ the late reverberation component, and t_{e} is the mixing time. The component $\mathbf{s}_{\mathrm{e}}(t)$ comprises the main peak of the RIR and the early reflections within a delay t_e which contribute to speech quality and intelligibility. The component $\mathbf{s}_{\mathrm{l}}(t)$ comprises all the later reflections which degrade intelligibility. The signals are transformed into the time-frequency domain by the short-time Fourier transform (STFT). In this domain, the recorded signal can be expressed as

$$\mathbf{d}(n,f) = \mathbf{s}(n,f) + \mathbf{y}(n,f) + \mathbf{b}(n,f) \tag{5}$$

$$= \mathbf{s}_{\mathrm{e}}(n,f) + \mathbf{s}_{\mathrm{l}}(n,f) + \mathbf{y}(n,f) + \mathbf{b}(n,f) \tag{6}$$

at time frame index $n \in [0, N-1]$ and frequency bin index $f \in [0, F-1]$, where F is the number of frequency bins and N the number of frames of the utterance. Note that the RIRs $\mathbf{a}_{\mathrm{y}}(\tau)$ and $\mathbf{a}_{\mathrm{s}}(\tau)$ are longer than the length of the STFT window. The goal is to recover the early near-end component $\mathbf{s}_{\mathrm{e}}(n,f) \in \mathbb{C}^{M\times 1}$ from the mixture $\mathbf{d}(n,f) \in \mathbb{C}^{M\times 1}$.

1.2 Model

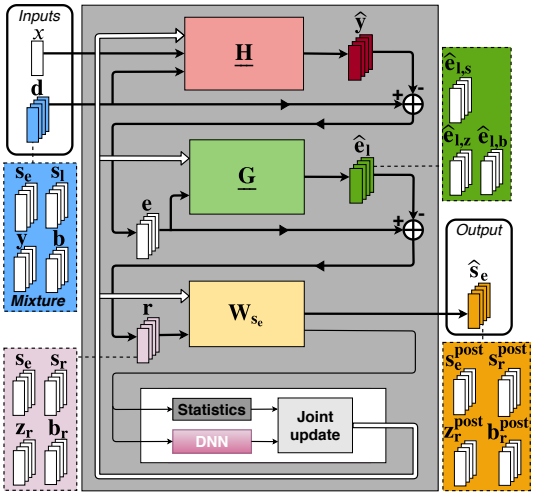


Figure 2: Proposed approach for joint reduction of echo, reverberation and noise. The bold arrows denote the filtering steps. The dashed lines denote the latent signal components. The thin arrows denote the signals used for the joint update. The white arrows denote the filter updates.

We propose a joint approach combining a linear echo cancellation filter $\underline{\mathbf{H}}(f)$, a linear dereverberation filter $\underline{\mathbf{G}}(f)$, and a nonlinear multichannel Wiener postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$. The approach is illustrated in Fig. 2. In the first step, we apply the echo cancellation filter $\underline{\mathbf{H}}(f) = [\mathbf{h}(0,f) \dots \mathbf{h}(K-1,f)] \in \mathbb{C}^{M\times K}$ on the K previous frames of the far-end signal $x(n,f)$, and subtract the resulting signal $\widehat{\mathbf{y}}(n,f)$ from $\mathbf{d}(n,f)$:

$$\mathbf{e}(n,f) = \mathbf{d}(n,f) - \underbrace{\sum_{k=0}^{K-1} \mathbf{h}(k,f) x(n-k,f)}_{=\widehat{\mathbf{y}}(n,f)} \tag{7}$$

where $\mathbf{h}(k,f) \in \mathbb{C}^{M\times 1}$ is the M-dimensional vector corresponding to the k-th tap of $\underline{\mathbf{H}}(f)$. Note

that in the multiframe filtering context, the tap k is measured in frames and the underscore notation in $\underline{\mathbf{H}}(f)$ denotes the concatenation of the K taps of $\mathbf{h}(k, f)$. The resulting signal $\mathbf{e}(n, f)$ contains the near-end signal $\mathbf{s}(n, f)$, the residual echo $\mathbf{z}(n, f)$ and the noise signal $\mathbf{b}(n, f)$:

$$\mathbf{e}(n, f) = \mathbf{s}(n, f) + \mathbf{b}(n, f) + \underbrace{\mathbf{y}(n, f) - \widehat{\mathbf{y}}(n, f)}_{=\mathbf{z}(n,f)}. \tag{8}$$

We then apply the dereverberation filter $\underline{\mathbf{G}}(f) = [\mathbf{G}(\Delta, f) \dots \mathbf{G}(\Delta + L - 1, f)] \in \mathbb{C}^{M \times ML}$ on the L previous frames of the signal $\mathbf{e}(n - \Delta, f)$ and subtract the resulting signal $\widehat{\mathbf{e}}_{\mathrm{l}}(n, f)$ from $\mathbf{e}(n, f)$. The resulting signal $\mathbf{r}(n, f)$ after reverberation filtering is thus

$$\mathbf{r}(n, f) = \mathbf{e}(n, f) - \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l, f)\mathbf{e}(n - l, f)}_{=\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)}, \tag{9}$$

where $\mathbf{G}(l, f) = [\mathbf{g}_1(l, f) \dots \mathbf{g}_M(l, f)] \in \mathbb{C}^{M \times M}$ is the $M \times M$-dimensional matrix corresponding to the l-th tap of $\underline{\mathbf{G}}(f)$. Since the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are applied on the previous frames of the signals $\mathbf{d}(n, f)$ and $x(n, f)$, we make the assumption that the observed signals $\mathbf{d}(n, f)$ and $x(n, f)$ are equal to zero for $n < 0$. Besides the target early near-end signal component $s_{\mathrm{e}}(n, f)$, the signal $\mathbf{r}(n, f)$ comprises three residual components: the residual near-end late reverberation $\mathbf{s}_{\mathrm{r}}(n, f)$, the *dereverberated* residual echo $\mathbf{z}_{\mathrm{r}}(n, f)$, and the *dereverberated* noise $\mathbf{b}_{\mathrm{r}}(n, f)$:

$$\mathbf{r}(n, f) = \mathbf{s}_{\mathrm{e}}(n, f) + \mathbf{s}_{\mathrm{r}}(n, f) + \mathbf{z}_{\mathrm{r}}(n, f) + \mathbf{b}_{\mathrm{r}}(n, f). \tag{10}$$

The term *dereverberated* means "after applying the dereverberation filter". The three residual signals can be expressed as

$$\mathbf{s}_{\mathrm{r}}(n, f) = \mathbf{s}_{\mathrm{l}}(n, f) - \widehat{\mathbf{e}}_{\mathrm{l},s}(n, f), \tag{11}$$

$$\mathbf{z}_{\mathrm{r}}(n, f) = \widehat{\mathbf{z}}(n, f) - \widehat{\mathbf{e}}_{\mathrm{l},z}(n, f), \tag{12}$$

$$\mathbf{b}_{\mathrm{r}}(n, f) = \mathbf{b}(n, f) - \widehat{\mathbf{e}}_{\mathrm{l},b}(n, f), \tag{13}$$

where the signals $\widehat{\mathbf{e}}_{\mathrm{l},s}(n, f)$, $\widehat{\mathbf{e}}_{\mathrm{l},y}(n, f)$ and $\widehat{\mathbf{e}}_{\mathrm{l},b}(n, f)$ are the latent components of $\widehat{\mathbf{e}}_{\mathrm{l}}(n, f)$ resulting

from (9)

$$\widehat{\mathbf{e}}_{\mathrm{l}}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\Big(\mathbf{s}(n-l,f) + \mathbf{z}(n-l,f) + \mathbf{b}(n-l,f)\Big) \tag{14}$$

$$= \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{s}(n-l,f)}_{=\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f)} + \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{z}(n-l,f)}_{=\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)} + \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{b}(n-l,f)}_{=\widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)}. \tag{15}$$

To recover $\mathbf{s}_{\mathrm{e}}(n,f)$ from $\mathbf{r}(n,f)$, we apply a multichannel Wiener postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f) \in \mathbb{C}^{M\times M}$:

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f) = \mathbf{W}_{s_{\mathrm{e}}}(n,f)\mathbf{r}(n,f). \tag{16}$$

Inspired by the weighted prediction error (WPE) method for dereverberation [2], we estimate $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ in the maximum likelihood (ML) sense by modeling the target $\mathbf{s}_{\mathrm{e}}(n,f)$ and the three residual signals $\mathbf{s}_{\mathrm{r}}(n,f)$, $\mathbf{z}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ with a multichannel local Gaussian framework. We thus consider each of these four signals as *sources* to be separated and model them as

$$\mathbf{s}_{\mathrm{e}}(n,f) \sim \mathcal{N}\Big(\mathbf{0}, v_{s_{\mathrm{e}}}(n,f)\mathbf{R}_{s_{\mathrm{e}}}(f)\Big), \tag{17}$$

$$\mathbf{s}_{\mathrm{r}}(n,f) \sim \mathcal{N}\Big(\mathbf{0}, v_{s_{\mathrm{r}}}(n,f)\mathbf{R}_{s_{\mathrm{r}}}(f)\Big), \tag{18}$$

$$\mathbf{z}_{\mathrm{r}}(n,f) \sim \mathcal{N}\Big(\mathbf{0}, v_{z_{\mathrm{r}}}(n,f)\mathbf{R}_{z_{\mathrm{r}}}(f)\Big), \tag{19}$$

$$\mathbf{b}_{\mathrm{r}}(n,f) \sim \mathcal{N}\Big(\mathbf{0}, v_{b_{\mathrm{r}}}(n,f)\mathbf{R}_{b_{\mathrm{r}}}(f)\Big), \tag{20}$$

where $v_{s_{\mathrm{e}}}(n,f) \in \mathbb{R}_+$ and $\mathbf{R}_{s_{\mathrm{e}}}(f) \in \mathbb{C}^{M\times M}$ denote the power spectral density (PSD) and the spatial covariance matrix (SCM) of the target $\mathbf{s}_{\mathrm{e}}(n,f)$, respectively. Note that $v_{s_{\mathrm{e}}}(n,f)$ is a time-varying nonnegative scalar and $\mathbf{R}_{s_{\mathrm{e}}}(f)$ is a full-rank Hermitian time-invariant matrix encoding the spatial properties of the target $\mathbf{s}_{\mathrm{e}}(n,f)$, since the conditions are time-invariant. Similarly, $v_{s_{\mathrm{r}}}(n,f)$ and $\mathbf{R}_{s_{\mathrm{r}}}(f)$ denote the PSD and SCM of the residual near-end reverberation $\mathbf{s}_{\mathrm{r}}(n,f)$, respectively, $v_{z_{\mathrm{r}}}(n,f)$ and $\mathbf{R}_{z_{\mathrm{r}}}(f)$ denote the PSD and SCM of the *dereverberated* residual echo $\mathbf{z}_{\mathrm{r}}(n,f)$, respectively, and $v_{b_{\mathrm{r}}}(n,f)$ and $\mathbf{R}_{b_{\mathrm{r}}}(f)$ denote the PSD and SCM of the *dereverberated* noise $\mathbf{s}_{\mathrm{r}}(n,f)$, respectively. The multichannel Wiener filter for the target $\mathbf{s}_{\mathrm{e}}(n,f)$ is thus

formulated as

$$\mathbf{W}_{s_\mathrm{e}}(n,f) = v_{s_\mathrm{e}}(n,f)\mathbf{R}_{s_\mathrm{e}}(f)\left(\sum_{c'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{21}$$

where the sum over c' includes all four sources. Similarly, the multichannel Wiener filters for the residual signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ are formulated as

$$\mathbf{W}_{s_\mathrm{r}}(n,f) = v_{s_\mathrm{r}}(n,f)\mathbf{R}_{s_\mathrm{r}}(f)\left(\sum_{c'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{22}$$

$$\mathbf{W}_{z_\mathrm{r}}(n,f) = v_{z_\mathrm{r}}(n,f)\mathbf{R}_{z_\mathrm{r}}(f)\left(\sum_{c'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{23}$$

$$\mathbf{W}_{b_\mathrm{r}}(n,f) = v_{b_\mathrm{r}}(n,f)\mathbf{R}_{b_\mathrm{r}}(f)\left(\sum_{c'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{24}$$

respectively.

1.3 Likelihood

In order to estimate the parameters of this model, we must first express its likelihood. Given its past sequence, and the current observation and past sequence of the far-end signal $x(n,f)$, the mixture signal $\mathbf{d}(n,f)$ is conditionally distributed as

$$\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{d}(n,f);\boldsymbol{\mu}_d(n,f),\mathbf{R}_d(n,f)\Big), \tag{25}$$

where

$$\boldsymbol{\mu}_{\mathbf{d}}(n,f) = \sum_{k=0}^{K-1}\mathbf{h}(k,f)x(n-k,f) + \sum_{l=\Delta}^{\Delta+L-1}\mathbf{G}(l,f)\mathbf{e}(n-l,f) \tag{26}$$

$$\mathbf{R}_{\mathbf{dd}}(n,f) = \sum_{c'} v_{c'}(n,f)\mathbf{R}_{c'}(n,f). \tag{27}$$

This is valid because $\mathbf{e}(n-l,f)$ is a linear combination of the observed mixture signal $\mathbf{d}(n-l,f)$, the far-end signal $x(n,f)$ and its past sequence (see (7)). The parameters to be estimated are

denoted by

$$\Theta_H = \{\underline{\mathbf{H}}(f)\}_f \tag{28}$$

$$\Theta_G = \{\underline{\mathbf{G}}(f)\}_f, \tag{29}$$

$$\Theta_c = \Big\{v_{s_\mathrm{e}}(n,f), \mathbf{R}_{s_\mathrm{e}}(f), v_{s_\mathrm{r}}(n,f), \mathbf{R}_{s_\mathrm{r}}(f), v_{z_\mathrm{r}}(n,f), \mathbf{R}_{z_\mathrm{r}}(f), v_{b_\mathrm{r}}(n,f), \mathbf{R}_{b_\mathrm{r}}(f)\Big\}_{n,f}. \tag{30}$$

The log-likelihood of the observed sequence $\mathcal{O} = \Big\{\mathbf{d}(n,f), x(n,f)\Big\}_{n,f}$ is then defined as

$$\mathcal{L}\Big(\mathcal{O}; \Theta_H, \Theta_G, \Theta_c\Big) = \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log p\Big(\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f), \ldots, \mathbf{d}(0,f), x(n,f), \ldots, x(0,f)\Big), \tag{31}$$

$$= \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log \mathcal{N}_{\mathbb{C}}\Big(\mathbf{d}(n,f); \boldsymbol{\mu}_{\mathbf{d}}(n,f), \mathbf{R}_{\mathbf{dd}}(n,f)\Big) \tag{32}$$

$$\equiv \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log |\mathbf{R}_{\mathbf{dd}}(n,f)|^{-1} - \Big(\mathbf{d}(n,f) - \boldsymbol{\mu}_{\mathbf{d}}(n,f)\Big)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\Big(\mathbf{d}(n,f) - \boldsymbol{\mu}_{\mathbf{d}}(n,f)\Big). \tag{33}$$

2 Vectorized computation of echo cancellation and dereverberation

As the filtering context is multiframe and multichannel, the resulting ML optimization problem is not separable across channels and taps. In order to solve it, the term $\widehat{\mathbf{y}}(n,f) = \sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f)$ is reformulated as

$$\widehat{\mathbf{y}}(n,f) = \underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f), \tag{34}$$

where $\underline{\mathbf{h}}(f) \in \mathbb{C}^{MK\times 1}$ is a vectorized version of the filter $\underline{\mathbf{H}}(f)$ as

$$\underline{\mathbf{h}}(f) = \begin{bmatrix} \mathbf{h}(0,f) \\ \vdots \\ \mathbf{h}(K-1,f) \end{bmatrix}, \tag{35}$$

$\underline{\mathbf{X}}(n,f) \in \mathbb{C}^{M \times MK}$ is the concatenation of the K taps of $\mathbf{X}(n-k,f) \in \mathbb{C}^{M \times M}$ as

$$\underline{\mathbf{X}}(n,f) = \Big[\mathbf{X}(n,f) \dots \mathbf{X}(n-K+1,f)\Big], \tag{36}$$

and $\mathbf{X}(n-k,f)$ is the multichannel version of $x(n-k,f)$ obtained as

$$\mathbf{X}(n-k,f) = x(n-k,f)\mathbf{I}_M. \tag{37}$$

Similarly, the term $\widehat{\mathbf{e}}_{\text{l}}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{e}(n-l,f)$ is reformulated as

$$\widehat{\mathbf{e}}_{\text{l}}(n,f) = \underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f), \tag{38}$$

where $\underline{\mathbf{g}}(f) \in \mathbb{C}^{M^2L \times 1}$ is a vectorized version of the filter $\underline{\mathbf{G}}(f)$ as [3]

$$\underline{\mathbf{G}}(f) = \begin{bmatrix} \\ \mathbf{g}_1(\Delta,f) \\ \vdots \\ \mathbf{g}_M(\Delta,f) \\ \vdots \\ \vdots \\ \mathbf{g}_1(\Delta+L-1,f) \\ \vdots \\ \mathbf{g}_M(\Delta+L-1,f) \end{bmatrix}, \tag{39}$$

$\underline{\mathbf{E}}(n,f) \in \mathbb{C}^{M \times M^2L}$ is the concatenation of the L taps $\mathbf{E}(n-l,f) \in \mathbb{C}^{M \times M^2}$ obtained as [3]

$$\underline{\mathbf{E}}(n,f) = \left[\mathbf{E}(n-\Delta,f) \dots \mathbf{E}(n-\Delta-L+1,f)\right], \tag{40}$$

and $\mathbf{E}(n-l,f)$ is the multichannel version of $\mathbf{e}(n-l,f)$ obtained as

$$\mathbf{E}(n-l,f) = \mathbf{I}_M \otimes \mathbf{e}(n-l,f)^T. \tag{41}$$

The resulting ML optimization problem has no closed-form solution, hence we need to estimate the parameters via an iterative procedure.

3 Iterative optimization algorithm

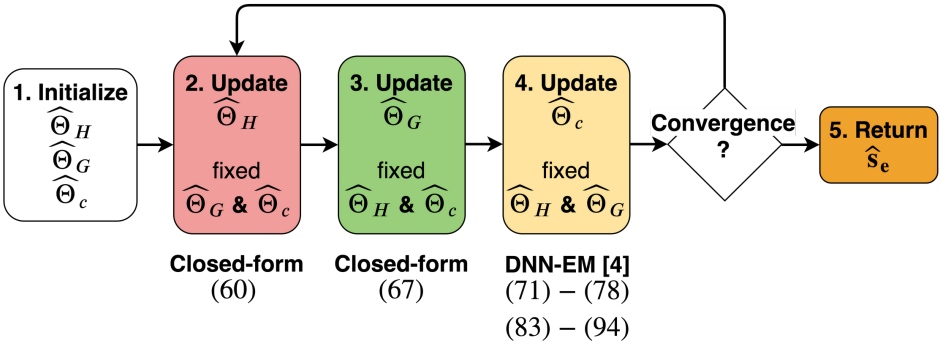


Figure 3: Flowchart of the proposed DNN-based BCA algorithm for likelihood optimization.

We propose a block-coordinate ascent (BCA) algorithm for likelihood optimization. Each iteration i comprises the following three maximization steps:

$$\widehat{\Theta}_H \leftarrow \underset{\Theta_H}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \Theta_H, \widehat{\Theta}_G, \widehat{\Theta}_c\right) \tag{42}$$

$$\widehat{\Theta}_G \leftarrow \underset{\Theta_G}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \Theta_G, \widehat{\Theta}_c\right) \tag{43}$$

$$\widehat{\Theta}_c \leftarrow \underset{\Theta_c}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \widehat{\Theta}_G.\Theta_c\right) \tag{44}$$

The solutions of (42) and (43) are closed-form. As there is no closed-form solution for (44), we propose to use Nugraha et al.'s DNN-EM algorithm [4]. The overall flowchart of the proposed algorithm is shown in Fig. 3. Note that it is also possible to optimize the parameters Θ_H, Θ_G and Θ_c with the EM algorithm by adding a noise term to (10) [5]. However, this approach would be less efficient to derive the filter parameters Θ_H and Θ_G. In the next subsections, we derive the update rules for the steps (42)–(44) of our proposed algorithm at iteration i.

3.1 Initialization

We initialize the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ to $\underline{\mathbf{H}}_0(f)$ and $\underline{\mathbf{G}}_0(f)$, respectively. The PSDs $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{e}}(n,f)$ and $v_{s_\mathrm{e}}(n,f)$ of the target and residual signals are jointly initialized using a pretrained DNN denoted as DNN_0 and the SCMs $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$ as the identity matrix $\mathbf{I}_M$. The inputs, the targets and the architecture of DNN_0 are described in Section IV below.

3.2 Echo cancellation filter parameters Θ_H

The partial derivative of the log-likelihood with respect to $\underline{\mathbf{h}}(f)$ can be computed as

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg) \tag{45}$$

$$= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \Big(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},z}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\Big)\Bigg)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \Big(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},z}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\Big)\Bigg) \tag{46}$$

Similarly to (14)–(15), we can split the term $\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)$ as

$$\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{z}(n-l,f) \tag{47}$$

$$= \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\Big(\mathbf{y}(n-l,f) - \widehat{\mathbf{y}}(n-k,f)\Big) \tag{48}$$

$$= \widehat{\mathbf{e}}_{\mathrm{l},y}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f), \tag{49}$$

and we replace it in (46):

$$\begin{aligned}
&\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} \\
&= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \left(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},y}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\right)\right)^H \\
&\qquad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\left(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \left(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},y}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f) + \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\right)\right)
\end{aligned} \tag{50}$$

Thus in (50), we can group the terms related to the signals $\mathbf{s}(n,f)$, $\mathbf{y}(n,f)$ and $\mathbf{b}(n,f)$ as they do not depend on $\underline{\mathbf{h}}(f)$:

$$\begin{aligned}
&\mathbf{d}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},y}(n,f) \\
&\qquad = \mathbf{d}(n,f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\Big(\mathbf{s}(n-l,f) + \mathbf{b}(n-l,f) + \mathbf{y}(n-l,f)\Big) && (51)\\
&\qquad = \mathbf{d}(n,f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{d}(n-l,f) && (52)\\
&\qquad = \mathbf{r}_d(n,f), && (53)
\end{aligned}$$

where $\mathbf{r}_d(n,f)$ is the *dereverberated* latent component of $\mathbf{r}(n,f)$ obtained by applying the dereverberation filter $\underline{\mathbf{G}}(f)$ on the mixture signal $\mathbf{d}(n,f)$ without prior echo cancellation. (50) becomes

$$\begin{aligned}
\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} &= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{r}_d(n,f) - \left(\widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f)\right)\right)^H \\
&\qquad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\left(\mathbf{r}_d(n,f) - \left(\widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f)\right)\right) && (54)\\
&= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{r}_d(n,f) - \left(\widehat{\mathbf{y}}(n,f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\widehat{\mathbf{y}}(n-l,f)\right)\right)^H \\
&\qquad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\left(\mathbf{r}_d(n,f) - \left(\widehat{\mathbf{y}}(n,f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\widehat{\mathbf{y}}(n-k,f)\right)\right). && (55)
\end{aligned}$$

Replacing (34) in (55):

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{r}_d(n,f) - \left(\underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\underline{\mathbf{X}}(n-l,f)\underline{\mathbf{h}}(f)\right)\right)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \left(\mathbf{r}_d(n,f) - \left(\underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f) - \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\underline{\mathbf{X}}(n-l,f)\underline{\mathbf{h}}(f)\right)\right). \tag{56}$$

$$= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{r}_d(n,f) - \underline{\mathbf{X}}_{\mathrm{r}}(n,f)\underline{\mathbf{h}}(f)\right)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \left(\mathbf{r}_d(n,f) - \underline{\mathbf{X}}_{\mathrm{r}}(n,f)\underline{\mathbf{h}}(f)\right), \tag{57}$$

where $\underline{\mathbf{X}}_{\mathrm{r}}(n,f) = [\mathbf{X}_{\mathrm{r}}(n,f)\dots\mathbf{X}_{\mathrm{r}}(n-K+1,f)] \in \mathbb{C}^{M\times MK}$ is the concatenation of the K taps $\mathbf{X}_{\mathrm{r}}(n,f) \in \mathbb{C}^{M\times M}$ which are *dereverberated* versions of $\mathbf{X}(n,f)$ obtained by applying the dereverberation filter $\underline{\mathbf{G}}(f)$ on the L previous frames of the far-end signal $x(n-k-l,f)$ and by subtracting the resulting signal from $\mathbf{X}(n,f)$ as

$$\mathbf{X}_{\mathrm{r}}(n,f) = \mathbf{X}(n,f) - \sum_{l=\Delta}^{\Delta+L-1} x(n-k-l,f)\mathbf{G}(l,f). \tag{58}$$

Thus (57) becomes

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} = \sum_{n=0}^{N-1} 2\underline{\mathbf{X}}_{\mathrm{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\underline{\mathbf{X}}_{\mathrm{r}}(n,f)\underline{\mathbf{h}}(f) - 2\underline{\mathbf{X}}_{\mathrm{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\mathbf{r}_d(n,f) \tag{59}$$

The log-likelihood is maximized with respect to $\underline{\mathbf{h}}(f)$ for $\frac{\partial \mathcal{L}(\mathcal{O};\Theta_H,\Theta_G,\Theta_c)}{\partial \underline{\mathbf{h}}(f)} = 0$. The echo cancellation filter $\underline{\mathbf{H}}(f)$ is thus updated as

$$\underline{\mathbf{h}}(f) = \mathbf{P}(f)^{-1}\mathbf{p}(f), \tag{60}$$

where

$$\mathbf{P}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}_{\mathrm{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{X}}_{\mathrm{r}}(n,f) \tag{61}$$

$$\mathbf{p}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}_{\mathrm{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \mathbf{r}_d(n,f) \tag{62}$$

Note that the matrix $\mathbf{P}(f)$ is a sum of rank-M terms, thus requires at least K terms in order to be invertible. The update of the echo cancellation filter $\underline{\mathbf{H}}(f)$ is influenced by the dereverberation filter $\underline{\mathbf{G}}(f)$ through the terms $\underline{\mathbf{X}}_{\mathrm{r}}(n,f)$ and $\mathbf{r}_d(n,f)$.

3.3 Dereverberation filter parameters Θ_G

The partial derivative of the log-likelihood with respect to $\underline{\mathbf{G}}(f)$ can be expressed as

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{g}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{g}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg)^H \\ \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg)\end{aligned} \tag{63}$$

The term $\mathbf{e}(n,f) = \mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f)$ does not depend on the linear dereverberation filter $\underline{\mathbf{g}}(\Delta,f)$, thus we obtain

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{g}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{g}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{e}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg)^H \\ \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \Bigg(\mathbf{e}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg).\end{aligned} \tag{64}$$

Replacing (38) in (64):

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{g}}(f)} &= -\frac{\partial}{\partial \underline{\mathbf{g}}(f)}\sum_{n=0}^{N-1}\left(\mathbf{e}(n,f)-\underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f)\right)^H \\ &\quad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\left(\mathbf{e}(n,f)-\underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f)\right) \qquad (65)\\ &= \sum_{n=0}^{N-1} 2\underline{\mathbf{E}}(n,f)^H\mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f) \\ &\quad - 2\underline{\mathbf{E}}(nf)^H\mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\mathbf{e}(n,f) \qquad (66)\end{aligned}$$

The log-likelihood is maximized with respect to $\underline{\mathbf{g}}(f)$ for $\frac{\partial \mathcal{L}(\mathcal{O};\Theta_H,\Theta_G,\Theta_c)}{\partial \underline{\mathbf{g}}(f)} = 0$. Similarly to WPE for dereverberation, the linear dereverberation filter $\underline{\mathbf{G}}(f)$ is thus updated as [6]

$$\underline{\mathbf{g}}(f) = \mathbf{Q}(f)^{-1}\mathbf{q}(f), \qquad (67)$$

where

$$\mathbf{Q}(f) = \sum_{n=0}^{N-1}\underline{\mathbf{E}}(n,f)^H\mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\underline{\mathbf{E}}(n,f) \qquad (68)$$

$$\mathbf{q}(f) = \sum_{n=0}^{N-1}\underline{\mathbf{E}}(n,f)^H\mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\mathbf{e}(n,f) \qquad (69)$$

Note that the matrix $\mathbf{Q}(f)$ is a sum of rank-M terms, thus requires at least ML terms in order to be invertible. The update of the dereverberation filter $\underline{\mathbf{G}}(f)$ is influenced by the echo cancellation filter $\underline{\mathbf{H}}(f)$ through the terms $\underline{\mathbf{E}}(n,f)$ and $\mathbf{e}(n,f)$.

3.4 Variance and spatial covariance parameters Θ_c

As there is no closed-form solution for the log-likelihood optimization with respect to Θ_c, the variance and spatial covariance parameters need to be estimated using an EM algorithm. Given the past sequence of the mixture signal $\mathbf{d}(n,f)$, the far-end signal $x(n,f)$ and its past sequence, and the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$, the residual mixture signal $\mathbf{r}(n,f)$ is conditionally distributed

as

$$\mathbf{r}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f),\underline{\mathbf{H}}(f),\underline{\mathbf{G}}(f)\sim\mathcal{N}_{\mathbb{C}}\Big(\mathbf{0},\mathbf{R}_{\mathbf{dd}}(n,f)\Big). \tag{70}$$

The signal model is conditionally identical to a multichannel local Gaussian modeling framework for source separation [7]. However, this framework does not constraint the PSDs or the SCMs which results in a permutation ambiguity in the separated components at each frequency bin f. Instead, after each update of the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$, we propose to use one iteration of Nugraha et al.'s DNN-EM algorithm to update the PSDs and the SCMs of the target and residual signals $\mathbf{s}_{\mathrm{e}}(n,f)$, $\mathbf{s}_{\mathrm{r}}(n,f)$, $\mathbf{z}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ [4]. In the E-step, the target and residual signals are estimated as

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f)=\mathbf{W}_{s_{\mathrm{e}}}(n,f)\mathbf{r}(n,f), \tag{71}$$

$$\widehat{\mathbf{s}}_{\mathrm{r}}(n,f)=\mathbf{W}_{s_{\mathrm{r}}}(n,f)\mathbf{r}(n,f), \tag{72}$$

$$\widehat{\mathbf{z}}_{\mathrm{r}}(n,f)=\mathbf{W}_{z_{\mathrm{r}}}(n,f)\mathbf{r}(n,f), \tag{73}$$

$$\widehat{\mathbf{b}}_{\mathrm{r}}(n,f)=\mathbf{W}_{b_{\mathrm{r}}}(n,f)\mathbf{r}(n,f), \tag{74}$$

and their second-order posterior moments as

$$\widehat{\mathbf{R}}_{s_{\mathrm{e}}}(n,f)=\widehat{\mathbf{s}}_{\mathrm{e}}(n,f)\widehat{\mathbf{s}}_{\mathrm{e}}(n,f)^{H}+\Big(\mathbf{I}-\mathbf{W}_{s_{\mathrm{e}}}(n,f)\Big)v_{s_{\mathrm{e}}}(n,f)\mathbf{R}_{s_{\mathrm{e}}}(f), \tag{75}$$

$$\widehat{\mathbf{R}}_{s_{\mathrm{r}}}(n,f)=\widehat{\mathbf{s}}_{\mathrm{r}}(n,f)\widehat{\mathbf{s}}_{\mathrm{r}}(n,f)^{H}+\Big(\mathbf{I}-\mathbf{W}_{s_{\mathrm{r}}}(n,f)\Big)v_{s_{\mathrm{r}}}(n,f)\mathbf{R}_{s_{\mathrm{r}}}(f), \tag{76}$$

$$\widehat{\mathbf{R}}_{z_{\mathrm{r}}}(n,f)=\widehat{\mathbf{z}}_{\mathrm{r}}(n,f)\widehat{\mathbf{z}}_{\mathrm{r}}(n,f)^{H}+\Big(\mathbf{I}-\mathbf{W}_{z_{\mathrm{r}}}(n,f)\Big)v_{z_{\mathrm{r}}}(n,f)\mathbf{R}_{z_{\mathrm{r}}}(f), \tag{77}$$

$$\widehat{\mathbf{R}}_{b_{\mathrm{r}}}(n,f)=\widehat{\mathbf{b}}_{\mathrm{r}}(n,f)\widehat{\mathbf{b}}_{\mathrm{r}}(n,f)^{H}+\Big(\mathbf{I}-\mathbf{W}_{b_{\mathrm{r}}}(n,f)\Big)v_{b_{\mathrm{r}}}(n,f)\mathbf{R}_{b_{\mathrm{r}}}(f). \tag{78}$$

In the M-step, the PSDs $v_{s_{\mathrm{e}}}(n,f)$, $v_{s_{\mathrm{r}}}(n,f)$, $v_{z\mathrm{r}}(n,f)$ and $v_{b_{\mathrm{r}}}(n,f)$ are jointly updated using a pretrained DNN denoted as DNN_i, with $i\geq 1$ the iteration index. The inputs, the targets and the architecture of DNN_i are described in Section IV below. In the exact EM algorithm, the

SCMs $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$ would be updated as [4]

$$\mathbf{R}_{s_\mathrm{e}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{s_\mathrm{e}}(n,f)}\widehat{\mathbf{R}}_{s_\mathrm{e}}(n,f), \tag{79}$$

$$\mathbf{R}_{s_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{s_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{s_\mathrm{r}}(n,f), \tag{80}$$

$$\mathbf{R}_{z_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{z_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{z_\mathrm{r}}(n,f), \tag{81}$$

$$\mathbf{R}_{b_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{b_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{b_\mathrm{r}}(n,f). \tag{82}$$

Here, for updating the SCMs $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$, we consider a weighted form of (80) instead [8]

$$\mathbf{R}_{s_\mathrm{e}}(f) = \left(\sum_{n=0}^{N-1} w_{s_\mathrm{e}}(n,f)\right)^{-1}\sum_{n=0}^{N-1}\frac{w_{s_\mathrm{e}}(n,f)}{v_{s_\mathrm{e}}(n,f)}\widehat{\mathbf{R}}_{s_\mathrm{e}}(n,f), \tag{83}$$

$$\mathbf{R}_{s_\mathrm{r}}(f) = \left(\sum_{n=0}^{N-1} w_{s_\mathrm{r}}(n,f)\right)^{-1}\sum_{n=0}^{N-1}\frac{w_{s_\mathrm{r}}(n,f)}{v_{s_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{s_\mathrm{r}}(n,f), \tag{84}$$

$$\mathbf{R}_{z_\mathrm{r}}(f) = \left(\sum_{n=0}^{N-1} w_{z_\mathrm{r}}(n,f)\right)^{-1}\sum_{n=0}^{N-1}\frac{w_{z_\mathrm{r}}(n,f)}{v_{z_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{z_\mathrm{r}}(n,f), \tag{85}$$

$$\mathbf{R}_{b_\mathrm{r}}(f) = \left(\sum_{n=0}^{N-1} w_{b_\mathrm{r}}(n,f)\right)^{-1}\sum_{n=0}^{N-1}\frac{w_{b_\mathrm{r}}(n,f)}{v_{b_\mathrm{r}}(n,f)}\widehat{\mathbf{R}}_{b_\mathrm{r}}(n,f), \tag{86}$$

where $w_{s_\mathrm{e}}(n,f)$, $w_{s_\mathrm{r}}(n,f)$, $w_{z_\mathrm{r}}(n,f)$ and $w_{b_\mathrm{r}}(n,f)$ denote the weight of the target and residual signals $\mathbf{s}_\mathrm{e}(n,f)$, $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$. When $w_{s_\mathrm{e}}(n,f) = w_{s_\mathrm{r}}(n,f) = w_{z_\mathrm{r}}(n,f) = w_{b_\mathrm{r}}(n,f) = 1$, (83) reduces to (80). Here, we use [8, 9]

$$w_{s_\mathrm{e}}(n,f) = v_{s_\mathrm{e}}(n,f), \tag{87}$$

$$w_{s_\mathrm{r}}(n,f) = v_{s_\mathrm{r}}(n,f), \tag{88}$$

$$w_{z_\mathrm{r}}(n,f) = v_{z_\mathrm{r}}(n,f), \tag{89}$$

$$w_{b_\mathrm{r}}(n,f) = v_{b_\mathrm{r}}(n,f). \tag{90}$$

Experience shows that this weighting trick mitigates inaccurate estimates in certain time-frequency bins and increases the importance of the bins for which $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{r}}(n,f)$, $v_{z_\mathrm{r}}(n,f)$ and $v_{b_\mathrm{r}}(n,f)$ are large. As the PSDs are constrained, we also need to constrain $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$ so as to encode only the spatial information of the sources. We modify (83) by normalizing $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$ after each update [8]:

$$\mathbf{R}_{s_\mathrm{e}}(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_{s_\mathrm{e}}(f)\right)}\mathbf{R}_{s_\mathrm{e}}(f), \tag{91}$$

$$\mathbf{R}_{s_\mathrm{r}}(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_{s_\mathrm{r}}(f)\right)}\mathbf{R}_{s_\mathrm{r}}(f), \tag{92}$$

$$\mathbf{R}_{z_\mathrm{r}}(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_{z_\mathrm{r}}(f)\right)}\mathbf{R}_{z_\mathrm{r}}(f), \tag{93}$$

$$\mathbf{R}_{b_\mathrm{r}}(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_{b_\mathrm{r}}(f)\right)}\mathbf{R}_{b_\mathrm{r}}(f). \tag{94}$$

3.5 Estimation of the final early near-end component $\mathbf{s_e}(n,f)$

Once the proposed iterative optimization algorithm has converged after I iterations, we have an estimation of the PSDs and the SCMs and we can ultimately derive the filters $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_\mathrm{e}}(n,f)$ to obtain the target estimate $\widehat{\mathbf{s}}_\mathrm{e}(n,f)$ using (7), (9) and (71). The detailed pseudo-code of the algorithm is provided in Alg. 1.

Algorithm 1: Proposed DNN-based BCA algorithm for joint reduction of echo, reverberation and noise

Input:
- $\mathbf{d}(n,f), x(n,f)$
- Pretrained $\text{DNN}_0, \text{DNN}_1, \ldots, \text{DNN}_I$

Initialize:
- Initialize the linear filters
  - $\underline{\mathbf{h}}(f) \leftarrow \underline{\mathbf{h}}_0(f)$ (chosen by the user, e.g. [10])
  - $\underline{\mathbf{g}}(f) \leftarrow \underline{\mathbf{g}}_0(f)$ (chosen by the user, e.g. WPE)
- Initialize the SCMs
  - $\left[\mathbf{R}_{s_\text{e}}(f)\ \mathbf{R}_{s_\text{r}}(f)\ \mathbf{R}_{z_\text{r}}(f)\ \mathbf{R}_{b_\text{r}}(f)\right] \leftarrow \left[\mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\right]$
- Initialize the DNN inputs
  - inputs $\leftarrow$ (112)
- Initialize the PSDs
  - $\left[v_{s_\text{e}}(n,f)\ v_{s_\text{r}}(n,f)\ v_{z_\text{r}}(n,f)\ v_{b_\text{r}}(n,f)\right] \leftarrow \text{DNN}_0(\text{inputs})$

for *each iteration* i *of* I **do**
- Update the echo cancellation filter
  - $\underline{\mathbf{h}}(f) \leftarrow$ (60)
- Update signal $\mathbf{e}(n,f)$
  - $\mathbf{e}(n,f) \leftarrow$ (7)
- Update the dereverberation filter
  - $\underline{\mathbf{g}}(f) \leftarrow$ (67)
- Update signal $\mathbf{r}(n,f)$
  - $\mathbf{r}(n,f) \leftarrow$ (9)
- Update the SCMs
- **for** *each spatial update* j *of* J **do**
  - **for** *each source* $\mathbf{c}$ *of* $[\mathbf{s}_\text{e}, \mathbf{s}_\text{r}, \mathbf{z}_\text{r}, \mathbf{b}_\text{r}]$ **do**
    - Update the multichannel Wiener filter
      - $\mathbf{W}_c(n,f) \leftarrow$ (21) or (22) or (23) or (24)
    - Update the source estimation
      - $\widehat{\mathbf{c}}(n,f) \leftarrow$ (71) or (72) or (73) or (74)
    - Update the posterior statistics
      - $\widehat{\mathbf{R}}_c(n,f) \leftarrow$ (75) or (76) or (77) or (78)
    - Update the source SCM
      - $\mathbf{R}_c(f) \leftarrow$ (83)+(87)+(91) or (84)+(88)+(92) or (85)+(89)+(93) or (86)+(90)+(94)
  - **end**
- **end**
- Update the DNN inputs
  - inputs $\leftarrow$ (114)
- Update the PSDs
  - $\left[v_{s_\text{e}}(n,f)\ v_{s_\text{r}}(n,f)\ v_{z_\text{r}}(n,f)\ v_{b_\text{r}}(n,f)\right] \leftarrow \text{DNN}_i(\text{inputs})$

end

Compute the final early near-end signal
- $\widehat{\mathbf{s}}_\text{e}(n,f) \leftarrow$ (7)+(9)+(71)

Output:
- $\widehat{\mathbf{s}}_\text{e}(n,f)$

4 DNN spectral model

In this section, we provide details for the derivation of inputs, targets and the architecture of the DNN used in the proposed BCA algorithm.

4.1 Targets

Estimating $\sqrt{v_c(n,f)}$ has been shown to provide better results than estimating the power spectra $v_c(n,f)$, as the square root compresses the signal dynamics [4]. Therefore we define $\left[\sqrt{v_{s_\mathrm{e}}(n,f)}\sqrt{v_{s_\mathrm{r}}(n,f)}\sqrt{v_{z_\mathrm{r}}(n,f)}\sqrt{v_{b_\mathrm{r}}(n,f)}\right]$ as the targets for the DNN. Nugraha et al. defined the ground truth PSD of any source $\mathbf{c}$ as [4]

$$v_c(n,f) = \frac{1}{M}\|\mathbf{c}(n,f)\|^2. \tag{95}$$

We thus need to know the ground truth source signals of the four sources. The ground truth latent signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ are unkown. However, in the training and validations sets, we can know the ground truth early near-end signal $\mathbf{s}_\mathrm{e}(n,f)$ and the signals $\mathbf{s}_\mathrm{l}(n,f)$, $\mathbf{y}(n,f)$ and $\mathbf{b}(n,f)$ [1]. These last three signals correspond to the values of the original distortion signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$, respectively, when the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are equal to zero. To derive the ground truth latent signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$, we thus propose to use an iterative algorithm similar to the BCA algorithm (see Fig. 3), where the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are initialized to zero. Therefore, at initialization, we set the three residual signals at the value of the original distortion signals:

$$\mathbf{s}_r(n,f) \leftarrow \mathbf{s}_\mathrm{l}(n,f), \tag{96}$$

$$\mathbf{z}_r(f) \leftarrow \mathbf{y}(n,f), \tag{97}$$

$$\mathbf{b}_r(f) \leftarrow \mathbf{b}(n,f). \tag{98}$$

We initialized the PSDs as

$$v_{s_\mathrm{e}}(n,f) \leftarrow \frac{1}{M}\|\mathbf{s}_\mathrm{e}(n,f)\|^2 \tag{99}$$

$$v_{s_\mathrm{r}}(n,f) \leftarrow \frac{1}{M}\|\mathbf{s}_\mathrm{r}(n,f)\|^2 \tag{100}$$

$$v_{z_\mathrm{r}}(n,f) \leftarrow \frac{1}{M}\|\mathbf{z}_\mathrm{r}(n,f)\|^2 \tag{101}$$

$$v_{b_\mathrm{r}}(n,f) \leftarrow \frac{1}{M}\|\mathbf{b}_\mathrm{r}(n,f)\|^2 \tag{102}$$

and the SCMs as the identity matrix $\mathbf{I}_M$. At each iteration, we derive the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ as in (60) and (67), respectively. Instead of using DNN-EM, we update $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ by applying the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ to each of the signals $\mathbf{s}_\mathrm{l}(n,f)$, $\mathbf{y}(n,f)$ and $\mathbf{b}(n,f)$ as in (11), (12) and (13). The detailed pseudo-code of the ground truth estimation procedure is provided in Alg. 2.

Algorithm 2: Proposed iterative procedure to derive the ground truths PSDs

Input:
$\mathbf{s}(n,f)$, $\mathbf{s}_{\mathrm{e}}(n,f)$, $\mathbf{s}_{\mathrm{l}}(n,f)$, $\mathbf{y}(n,f)$, $\mathbf{b}(n,f)$
Initialize:
Initialize the latent variables
$\mathbf{s}_r(n,f) \leftarrow \mathbf{s}_{\mathrm{l}}(n,f)$
$\mathbf{z}_r(f) \leftarrow \mathbf{y}(n,f)$
$\mathbf{b}_r(f) \leftarrow \mathbf{b}(n,f)$
for *each source* $\mathbf{c}$ *of* $[\mathbf{s}_{\mathrm{e}}, \mathbf{s}_{\mathrm{r}}, \mathbf{z}_{\mathrm{r}}, \mathbf{b}_{\mathrm{r}}]$ **do**
Initialize the PSDs
$v_c(n,f) \leftarrow$ (99) or (100) or (101) or (102)
Initialize the SCMs
$\mathbf{R}_c(f) \leftarrow \mathbf{I}_M$
end
for *each iteration* i *of* I **do**
Update the echo cancellation filter
$\underline{\mathbf{h}}(f) \leftarrow$ (60)
Update the dereverberation filter
$\underline{\mathbf{g}}(f) \leftarrow$ (67)
Update the latent variables
$\mathbf{s}_{\mathrm{r}}(n,f) \leftarrow$ (11)
$\mathbf{z}_{\mathrm{r}}(n,f) \leftarrow$ (12)
$\mathbf{b}_{\mathrm{r}}(n,f) \leftarrow$ (13)
for *each source* $\mathbf{c}$ *of* $[\mathbf{s}_{\mathrm{e}}, \mathbf{s}_{\mathrm{r}}, \mathbf{z}_{\mathrm{r}}, \mathbf{b}_{\mathrm{r}}]$ **do**
Update the source PSD
$v_c(n,f) \leftarrow$ (103) or (104) or (105) or (106)
Update the source SCM
$\mathbf{R}_c(f) \leftarrow$ (107)+(91) or (108)+(92) or (109)+(93) or (110)+(94)
end
end
Output:
$\left[v_{s_{\mathrm{e}}}(n,f)\ v_{s_{\mathrm{r}}}(n,f)\ v_{z_{\mathrm{r}}}(n,f)\ v_{b_{\mathrm{r}}}(n,f)\right]$

As we aim at obtaining the ground truth PSDs $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{r}}(n,f)$, $v_{z_\mathrm{r}}(n,f)$ and $v_{b_\mathrm{r}}(n,f)$, we propose to use one iteration of Duong et al.'s EM algorithm to update the PSDs and the SCMs of the target and residual signals $\mathbf{s}_\mathrm{e}(n,f)$, $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ separately [7]. A similar procedure was used by Nugraha et al. [8]. In the E-step, replacing $\widehat{\mathbf{s}}_\mathrm{e}(n,f)$, $\widehat{\mathbf{s}}_\mathrm{r}(n,f)$, $\widehat{\mathbf{z}}_\mathrm{r}(n,f)$ and $\widehat{\mathbf{b}}_\mathrm{r}(n,f)$ by $\mathbf{s}_\mathrm{e}(n,f)$, $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$, respectively, and $\widehat{\mathbf{R}}_{s_\mathrm{e}}(n,f)$, $\widehat{\mathbf{R}}_{s_\mathrm{r}}(n,f)$, $\widehat{\mathbf{R}}_{z_\mathrm{r}}(n,f)$ and $\widehat{\mathbf{R}}_{b_\mathrm{r}}(n,f)$ by $\mathbf{s}_\mathrm{e}(n,f)\mathbf{s}_\mathrm{e}(n,f)^H$, $\mathbf{s}_\mathrm{r}(n,f)\mathbf{s}_\mathrm{r}(n,f)^H$, $\mathbf{z}_\mathrm{r}(n,f)\mathbf{z}_\mathrm{r}(n,f)^H$ and $\mathbf{b}_\mathrm{r}(n,f)\mathbf{b}_\mathrm{r}(n,f)^H$, respectively, the PSDs are computed as

$$v_{s_\mathrm{e}}(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_{s_\mathrm{e}}^{-1}(f)\mathbf{s}_\mathrm{e}(n,f)\mathbf{s}_\mathrm{e}(n,f)^H\Big) \tag{103}$$

$$v_{s_\mathrm{r}}(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_{s_\mathrm{r}}^{-1}(f)\mathbf{s}_\mathrm{r}(n,f)\mathbf{s}_\mathrm{r}(n,f)^H\Big), \tag{104}$$

$$v_{z_\mathrm{r}}(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_{z_\mathrm{r}}^{-1}(f)\mathbf{z}_\mathrm{r}(n,f)\mathbf{z}_\mathrm{r}(n,f)^H\Big), \tag{105}$$

$$v_{b_\mathrm{r}}(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_{b_\mathrm{r}}^{-1}(f)\mathbf{b}_\mathrm{r}(n,f)\mathbf{b}_\mathrm{r}(n,f)^H\Big), \tag{106}$$

and the SCMs as

$$\mathbf{R}_{s_\mathrm{e}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{s_\mathrm{e}}(n,f)}\mathbf{s}_\mathrm{e}(n,f)\mathbf{s}_\mathrm{e}(n,f)^H, \tag{107}$$

$$\mathbf{R}_{s_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{s_\mathrm{r}}(n,f)}\mathbf{s}_\mathrm{r}(n,f)\mathbf{s}_\mathrm{r}(n,f)^H, \tag{108}$$

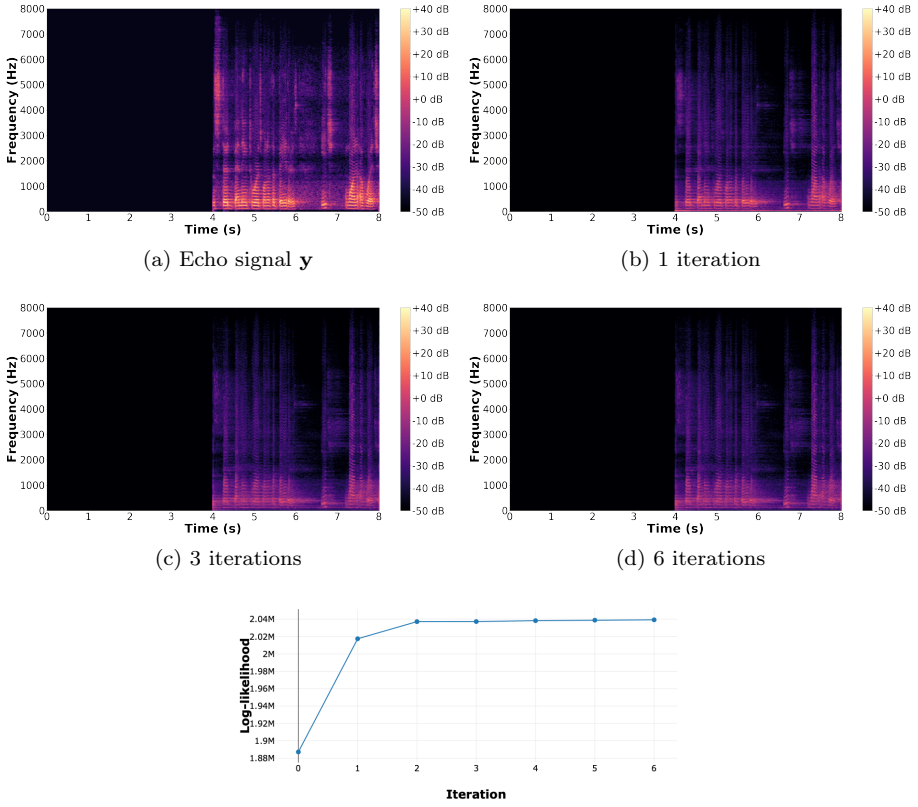
$$\mathbf{R}_{z_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{z_\mathrm{r}}(n,f)}\mathbf{z}_\mathrm{r}(n,f)\mathbf{z}_\mathrm{r}(n,f)^H, \tag{109}$$

$$\mathbf{R}_{b_\mathrm{r}}(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_{b_\mathrm{r}}(n,f)}\mathbf{b}_\mathrm{r}(n,f)\mathbf{b}_\mathrm{r}(n,f)^H. \tag{110}$$

The SCMs are then normalized so as to encode only the spatial information of the sources as in (91), (92), (93) and (94).

Note that we initialize the ground truth estimation procedure in a similar way as WPE for dereverberation [2]. Indeed, in WPE, the early near-end signal $\mathbf{s}_\mathrm{e}(n,f)$ is a latent variable and is initialized with the reverberant near-end signal as $\mathbf{s}_\mathrm{e}(n,f) \leftarrow \mathbf{s}(n,f)$. The initialization of Alg. 2 proved to provide significant reduction of echo and reverberation. After a few iterations, we observed the convergence of the estimated ground truth residual signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and

$\mathbf{b}_{\mathrm{r}}(n,f)$. In practice, we found that the signal $\mathbf{z}_{\mathrm{r}}(n,f)$ derived with this iterative procedure did not change after 3 iterations (see Fig. 4). The signals $\mathbf{s}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ did not change after 1 iteration.



(a) Echo signal $\mathbf{y}$

(b) 1 iteration

(c) 3 iterations

(d) 6 iterations

(e) Log-likelihood

Figure 4: Example estimates of the ground truth residual echo PSD $v_{z_{\mathrm{r}}}$ obtained by Alg. 2.

4.2 Inputs

We use magnitude spectra as inputs for DNN_0 and DNN_i rather than power spectra, since they have been shown to provide better results when the targets are the magnitude spectra $\sqrt{v_c(n,f)}$ [4]. The inputs of DNN_0 and DNN_i are summarized in Fig. 5. We concatenate these spectra to obtain the inputs. For DNN_0, we consider first the far-end signal magnitude $|x(n,f)|$ and a single-channel signal magnitude $|\widetilde{d}(n,f)|$ obtained from the corresponding multichannel

mixture signal $\mathbf{d}(n,f)$ as [8]

$$|\widetilde{d}(n,f)| = \sqrt{\frac{1}{M}\|\mathbf{d}(n,f)\|^2}. \tag{111}$$

Additionally we use the magnitude spectra of the signals $|\widetilde{y}(n,f)|$, $|\widetilde{e}(n,f)|$, $|\widetilde{e}_1(n,f)|$ and $|\widetilde{r}(n,f)|$ obtained from the corresponding multichannel signals after each linear filtering step $\widehat{\mathbf{y}}(n,f)$, $\mathbf{e}(n,f)$, $\widehat{\mathbf{e}}_1(n,f)$, $\mathbf{r}(n,f)$. Indeed in our previous work on single-channel echo reduction, using the estimated echo magnitude as an additional input was shown to improve the estimation [11]. We refer to the above inputs as type-I inputs. The inputs of DNN_0 are concatenated as

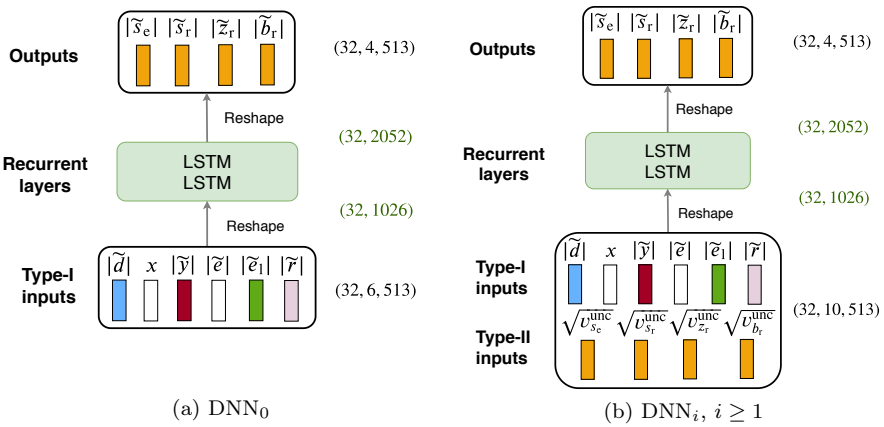
$$\text{inputs} = \left[|\widetilde{d}(n,f)|, |x(n,f)|, |\widetilde{y}(n,f)|, |\widetilde{e}(n,f)|, |\widetilde{e}_1(n,f)|, |\widetilde{r}(n,f)|\right]. \tag{112}$$

For DNN_i, we consider additional inputs to improve the DNN estimation. In particular, we use the magnitude spectra $\sqrt{v_c^{\text{unc}}(n,f))}$ of the source unconstrained PSDs obtained as

$$v_c^{\text{unc}}(n,f) = \frac{1}{M}\text{tr}\left(\mathbf{R}_c(f)^{-1}\widehat{\mathbf{R}}_c(n,f)\right). \tag{113}$$

Indeed these inputs partially contain the spatial information of the sources and have been shown to improve results in source separation [4]. We denote the inputs obtained from (113) as type-II inputs. The inputs of DNN_i are concatenated as

$$\text{inputs} = \left[|\widetilde{d}(n,f)|, |x(n,f)|, |\widetilde{y}(n,f)|, |\widetilde{e}(n,f)|, |\widetilde{e}_1(n,f)|, |\widetilde{r}(n,f)|, \left[\sqrt{v_{c'}^{\text{unc}}(n,f))}\right]_{c'}\right]. \tag{114}$$



(a) DNN_0 (b) DNN_i, $i \geq 1$

Figure 5: Architecture of the DNNs with a sequence length of 32 timesteps and $F = 513$ frequency bins.

5 Recording and simulation parameters

In this section we give details of the recording and simulation parameters for creating the datasets.

5.1 Overall description

We created three disjoint datasets for training, validation and test, whose characteristics are summarized in Table 1. For each dataset, we separately recorded or simulated the acoustic echo $\mathbf{y}(t)$, the near-end speech $\mathbf{s}(t)$ and the noise $\mathbf{b}(t)$ using clean speech and noise signals as base material and we computed the mixture signal $\mathbf{d}(t)$ as in (1). This protocol is required to obtain the ground truth target and residual signals for training and evaluation, which is not possible with real-world recordings for which these ground truth signals are unknown. The training and validation sets correspond to time-invariant acoustic conditions, while the test set includes both a time-invariant and a time-varying subset.

Real echo recordings In real hands-free systems, the acoustic echo contains nonlinearities caused by the nonlinear response of the loudspeaker, enclosure vibrations and hard clipping effects due to amplification. In order to achieve more realistic test conditions, we created the acoustic echo by recording the acoustic feedback from the loudspeaker to the microphones of a real hands-free system. The far-end speech was played and recorded at a rate of 16 kHz with a Triby, a

Dataset		**Training**	**Validation**	**Test**
Signals	**y**	recorded		recorded
	s	simulated $\mathbf{a}_s$		
	b	simulated		
Rooms		1-2-3	1-2	4
# speaker pairs		79	27	25
# utterances		13,572	4,536	4,500
# noise samples		36	36	6
SER range (dB)		$[-45, +6]$		$[-45, -7]$
SNR range (dB)		$[-21, +24]$		$[-20, +13]$

Table 1: Dataset characteristics.

smart speaker device developed by Invoxia which has a uniform linear array of 4 microphones. The distance between the loudspeaker (playing the far-end signal) and the microphones was 11 cm, and the distance between the microphones was 3 cm. One of the microphones exhibited some recording problem due to occlusion and was discarded. We only used the signals of $M = 3$ microphones. The recordings were done with the same Triby in 4 rooms with different size and reverberation time (RT_{60}) listed in Table 2. The smart speaker was placed on a table in the center of each room (see Fig. 6) at 2 differents positions to increase the diversity of the echo paths. The far-end speech was played by the loudspeaker at 3 different loudness levels to increase the diversity of nonlinearities: the louder the far-end speech, the larger the nonlinearities.

Room	**Size** (m)	$\mathbf{RT}_{60}$ (s)
1	$4.4 \times 4.2 \times 4$	1.0
2	$3.8 \times 2.5 \times 3.5$	0.5
3	$3.4 \times 2.1 \times 3.3$	0.8
4	$3.4 \times 2.1 \times 3.3$	1.3

Table 2: Room characteristics.

Real or simulated near-end speech and noise The creation procedures for $\mathbf{s}(t)$ and $\mathbf{b}(t)$ differ for each dataset and are described in the following subsections.

5.2 Training set

For the training set, the echo recordings were done in rooms 1, 2 and 3 (see Table 2). To create the reverberant near-end speech $\mathbf{s}(t)$, we convolved anechoic near-end speech $u(t)$ with near-end

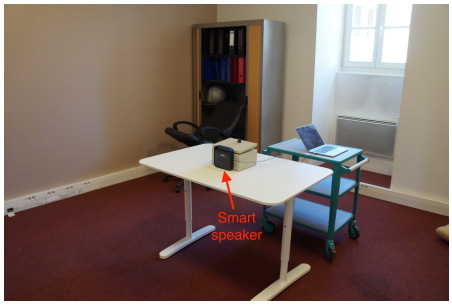


Figure 6: Echo recording setup.

RIRs $\mathbf{a}_s(\tau)$ simulated using the Roomsimove toolbox [12]. The simulated rooms matched the same reverberation characteristics (RT_{60} per octave) as the real rooms where the echo recordings were done (see Fig. 7). However the simulated room size was chosen randomly within $\pm 20\,\%$ of the real room size. The dimensions of the simulated microphone array were identical to those of the real Triby, and its simulated position and orientation were similar. The position of the near-end speaker was chosen randomly on a semicircle of 1.5 m radius centered in the middle of the microphone array and at least 10 cm from the walls (see Fig. 8). For each RIR, a random room and a random near-end position were generated to increase the diversity of RIRs.

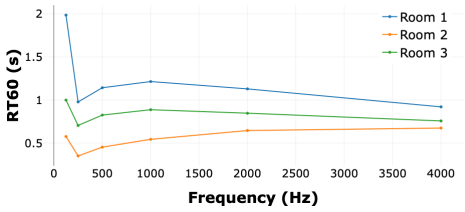


Figure 7: RT_{60} per octave of rooms 1, 2 and 3.

5.3 Validation set

The validation set was generated in a similar way as the training set, using 27 speaker pairs and 36 noise samples that are not in the training set. The echo recordings were done in rooms 1 and 2, and the near-end RIRs were simulated using the reverberation characteristics of these two rooms (see Fig. 7).

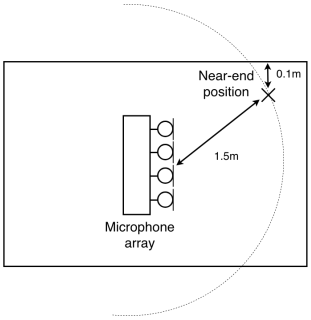


Figure 8: Near-end simulation settings.

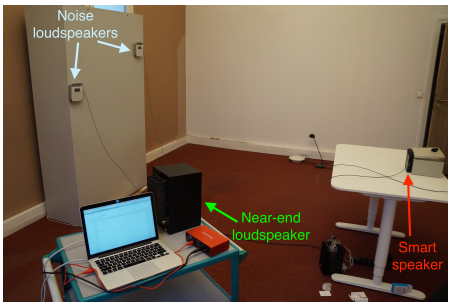


Figure 9: Recording setup for the test set.

5.4 Time-invariant test set

The time-invariant test set was built from real recordings only, using 25 speaker pairs and 6 noise samples that are neither in the training nor in the validation sets. The echo, the near-end speech and the noise were all recorded in room 4 (see Table 2) using the setup shown in Fig. 9. The reverberant near-end speech $\mathbf{s}(t)$ was obtained by playing anechoic speech with a Yamaha MSP5 Studio loudspeaker at a single loudness level. This loudspeaker was placed on a semicircle of 1.5 m radius centered in the middle of the smart speaker at 3 positions: 0°, 60°and −60°. The noise signal $\mathbf{b}(t)$ was obtained by randomly picking an original noise signal and playing it through 4 Triby loudspeakers simutaneously. These loudspeakers were placed at 2.5 m from the microphone array and their position remained fixed during all recordings (see Fig. 9). Similarly to the far-end speech, the noise signals were played at 3 different loudness levels. We considered a realistic use case scenario where the user increases the playback level of the smart speaker in the presence of a louder noise, hence louder noise signals were matched with louder far-end speech.

References

[1] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Joint DNN-based multichannel reduction of echo, reverberation and noise," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, submitted.

[2] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[3] T. Yoshioka, T. Nakatani, M. Miyoshi, and H. G. Okuno, "Blind separation and dereverberation of speech mixtures by joint optimization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 69–84, 2011.

[4] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[5] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[6] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[7] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.

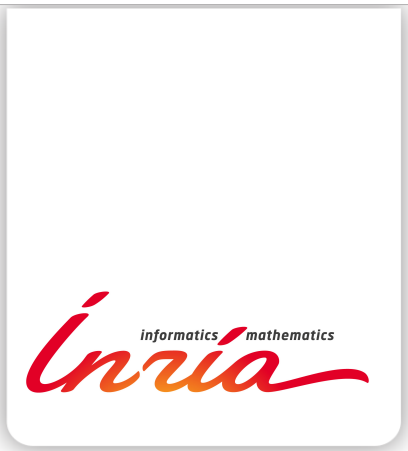
[8] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. EUSIPCO*, 2016, pp. 1748–1752.

[9] A. Liutkus, D. Fitzgerald, and Z. Rafii, "Scalable audio separation with light kernel additive modelling," in *Proc. ICASSP*, 2015, pp. 76–80.

[10] J. M. Valin, "On adjusting the learning rate in frequency domain echo cancellation with double-talk," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1030–1034, 2007.

[11] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Multiple-input neural network-based residual echo suppression," in *Proc. ICASSP*, 2018, pp. 231–235.

[12] E. Vincent and D. R. Campbell, "Roomsimove," 2008. [Online]. Available: http://homepages.loria.fr/evincent/software/Roomsimove_1.4.zip



RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique

CS20101

54603 Villers-lès-Nancy Cedex

Publisher

Inria

Domaine de Voluceau - Rocquencourt

BP 105 - 78153 Le Chesnay Cedex

inria.fr

ISSN 0249-6399