Joint NN-Supported Multichannel Reduction of Acoustic Echo, Reverberation and Noise: Supporting Document

Guillaume Carbajal, Romain Serizel, Emmanuel Vincent, Eric Humbert

Joint NN-Supported Multichannel Reduction of Acoustic Echo, Reverberation and Noise: Supporting Document

Guillaume Carbajal, Romain Serizel , Emmanuel Vincent , Éric Humbert

RESEARCH

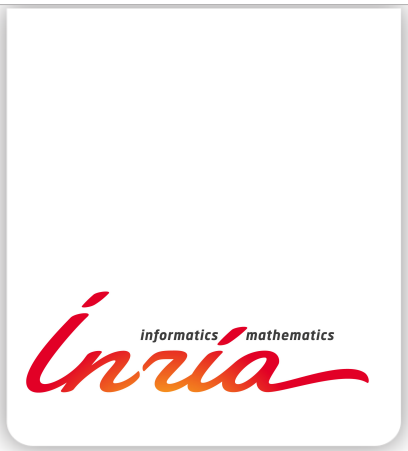
REPORT

N° 9303

November 2019

Project-Team Multispeech

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



Joint NN-Supported 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 — 71 pages

* 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

Abstract: This technical report is the supporting document of our proposed approach based on a neural network for joint multichannel reduction of echo, reverberation and noise [1]. First, we recall the model of the proposed approach. Secondly, we express the vectorized computation of echo cancellation and dereverberation. Thirdly, we detail the complete derivation of the update rules. Fourthly we describe the computation of the ground truth targets for the neural network used in our approach. Fifthly we detail the variant of the proposed approach where echo cancellation and dereverberation are performed in parallel. Sixthly we describe the variant of the proposed approach where only echo cancellation is performed. Then we specify the recording and simulation parameters of the dataset, we detail the computation of the estimated early near-end components and we recall the baselines. Finally we give the results after each filtering step and provides estimated spectrogram examples by all the approaches.

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

Contents

This technical report is organized as follows. In Section 1, we recall the model of the proposed approach. We express the vectorized computation of echo cancellation and dereverberation in Section 2. In section 3 we detail the complete derivation of the update rules. Section 4 describes the computation of the ground truth targets for the neural network used in our approach. In Section 5 we detail the variant of the proposed approach where echo cancellation and dereverberation are performed in parallel. Section 6 describes the variant of the proposed approach where only echo cancellation is performed. In Section 7 we specify the recording and simulation parameters of the dataset, we detail the computation of the estimated early near-end components and we recall the baselines. Finally Section 8 gives the results after each filtering step and provides estimated spectrogram examples by all the approaches.

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

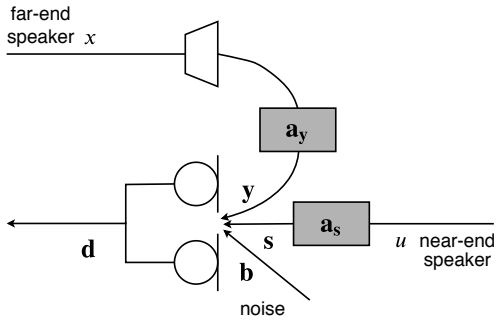


Figure 1: Acoustic echo, reverberation and noise problem.

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 echo signal can be expressed as

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

The linear part corresponds to 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 nonlinear part is denoted by $\mathbf{y}_{\mathrm{nl}}(t) \in \mathbb{R}^{M\times 1}$. 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

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

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

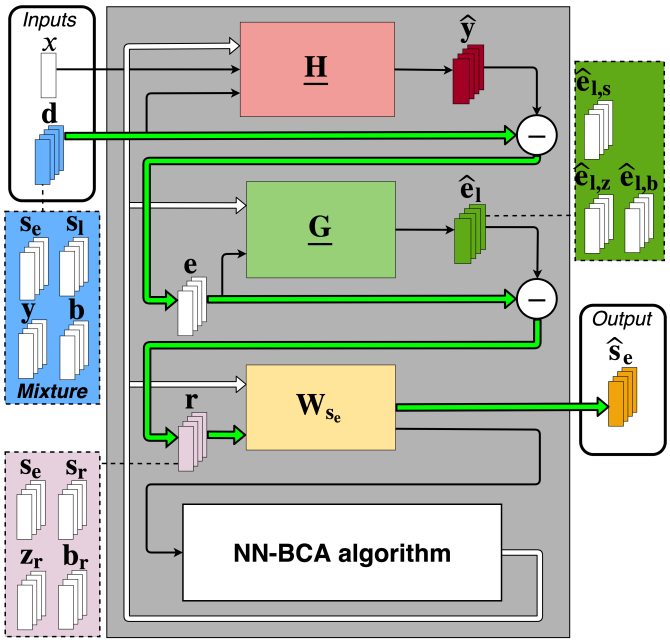


Figure 2: Proposed approach for joint reduction of echo, reverberation and noise. The bold green arrows denote the filtering steps. The dashed lines denote the latent signal components. The thin black arrows denote the signals used for the filtering steps and for the joint update. The bold white arrows 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 echo estimate $\widehat{\mathbf{y}}(n,f)$ 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

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

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

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

where the signals $\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f)$, $\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)$ 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

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

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

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

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

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

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

where $\mathcal{C} = \{\mathbf{s}_{\mathrm{e}}, \mathbf{s}_{\mathrm{r}}, \mathbf{z}_{\mathrm{r}}, \mathbf{b}_{\mathrm{r}}\}$ denotes all four sources in signal $\mathbf{r}(n,f)$. 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

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

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

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

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

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

where

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

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

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 $\mathbf{X}(n-k,f) \in \mathbb{C}^{M\times M}$ 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 $\widehat{\mathbf{e}}_{\mathrm{l}}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{e}(n-l,f)$ is reformulated as

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

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

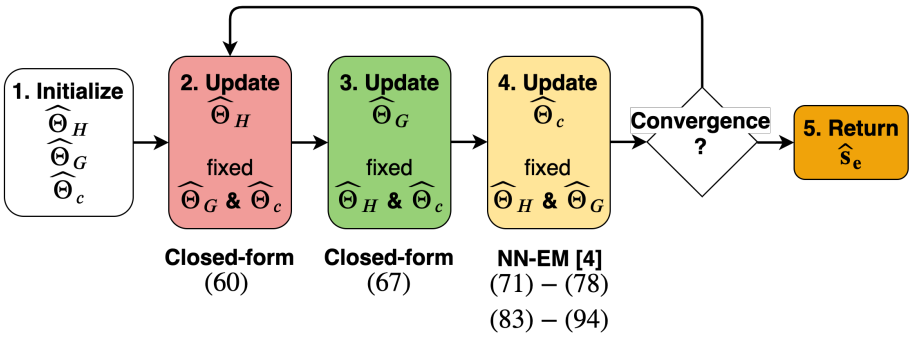


Figure 3: Flowchart of the proposed NN-supported BCA algorithm for likelihood optimization.

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 NN-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 $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{r}}(n,f)$, $v_{z_\mathrm{r}}(n,f)$ and $v_{b_\mathrm{r}}(n,f)$ of the target and residual signals are jointly initialized using a pretrained NN denoted as NN_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 NN_0 are described in Section 4 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

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} &= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)}\sum_{n=0}^{N-1}\Bigg(\mathbf{d}(n,f)-\widehat{\mathbf{y}}(n,f)-\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg)^H \\ &\quad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\Bigg(\mathbf{d}(n,f)-\widehat{\mathbf{y}}(n,f)-\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)\Bigg) \qquad (45)\\ &= -\frac{\partial}{\partial \underline{\mathbf{h}}(f)}\sum_{n=0}^{N-1}\Bigg(\mathbf{d}(n,f)-\widehat{\mathbf{y}}(n,f)-\Big(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f)+\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)+\widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\Big)\Bigg)^H \\ &\quad \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\Bigg(\mathbf{d}(n,f)-\widehat{\mathbf{y}}(n,f)-\Big(\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f)+\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)+\widehat{\mathbf{e}}_{\mathrm{l},b}(n,f)\Big)\Bigg). \qquad (46)\end{aligned}$$

Similarly to (120)–(121), we can split the term $\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f)$ as

$$\begin{aligned}\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f) &= \sum_{l=\Delta}^{\Delta+L-1}\mathbf{G}(l,f)\mathbf{z}(n-l,f) &(47)\\ &= \sum_{l=\Delta}^{\Delta+L-1}\mathbf{G}(l,f)\Big(\mathbf{y}(n-l,f)-\widehat{\mathbf{y}}(n-l,f)\Big) &(48)\\ &= \widehat{\mathbf{e}}_{\mathrm{l},y}(n,f)-\widehat{\mathbf{e}}_{\mathrm{l},\widehat{y}}(n,f), &(49)\end{aligned}$$

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 $\underline{\mathbf{h}}(f)$:

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

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

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

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 $\underline{\mathbf{g}}(f)$, 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):

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\mathbf{g}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{g}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{e}(n,f) - \underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f)\right)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \left(\mathbf{e}(n,f) - \underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f)\right) \tag{65}$$

$$= \sum_{n=0}^{N-1} 2\underline{\mathbf{E}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{E}}(n,f)\underline{\mathbf{g}}(f) - 2\underline{\mathbf{E}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1}\mathbf{e}(n,f). \tag{66}$$

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]

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

where

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

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

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 NN-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, if we used 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)–(86) 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]:

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

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

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

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

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 NN denoted as NN_i, with $i \geq 1$ the iteration index. The inputs, the targets and the architecture of NN_i are described in Section 4 below.

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 estimates of the PSDs $v_c(n,f)$, the SCMs $\mathbf{R}_c(f)$ and the dereverberation filter $\underline{\mathbf{G}}(f)$. We can perform one more iteration of the NN-supported BCA algorithm to derive the final filters $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_\mathrm{e}}(n,f)$. Ultimately, we obtain the target estimate $\widehat{\mathbf{s}}_\mathrm{e}(n,f)$ using (7), (9) and (16). For the detailed pseudo-code of the algorithm, please refer to the supporting document [10]. The detailed pseudo-code of the algorithm is provided in Alg. 1.

Algorithm 1: Proposed NN-supported BCA algorithm for joint reduction of echo, reverberation and noise.

Input:
$\mathbf{d}(n,f), x(n,f)$
Pretrained $\mathrm{NN}_0, \mathrm{NN}_1, \ldots, \mathrm{NN}_I$
Initialize:
Initialize the linear filters
$\underline{\mathbf{h}}(f) \leftarrow \underline{\mathbf{h}}_0(f)$ (chosen by the user, e.g. [11])
$\underline{\mathbf{g}}(f) \leftarrow \underline{\mathbf{g}}_0(f)$ (chosen by the user, e.g. WPE)
Initialize the SCMs
$\left[\mathbf{R}_{s_\mathrm{e}}(f)\ \mathbf{R}_{s_\mathrm{r}}(f)\ \mathbf{R}_{z_\mathrm{r}}(f)\ \mathbf{R}_{b_\mathrm{r}}(f)\right] \leftarrow \left[\mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\right]$
Initialize the NN inputs
inputs $\leftarrow$ (112)
Initialize the PSDs
$\left[v_{s_\mathrm{e}}(n,f)\ v_{s_\mathrm{r}}(n,f)\ v_{z_\mathrm{r}}(n,f)\ v_{b_\mathrm{r}}(n,f)\right] \leftarrow \left[\mathrm{NN}_0(\text{inputs})\right]^2$
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* $[\mathbf{s}_\mathrm{e}, \mathbf{s}_\mathrm{r}, \mathbf{z}_\mathrm{r}, \mathbf{b}_\mathrm{r}]$ **do**
      Update the multichannel Wiener filter
      $\mathbf{W}_c(n,f) \leftarrow$ (21) or (22) or (23) or (24)
      Update the source estimation
      $\widehat{\mathbf{c}}(n,f) \leftarrow$ (16) 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 NN inputs
  inputs $\leftarrow$ (114)
  Update the PSDs
  $\left[v_{s_\mathrm{e}}(n,f)\ v_{s_\mathrm{r}}(n,f)\ v_{z_\mathrm{r}}(n,f)\ v_{b_\mathrm{r}}(n,f)\right] \leftarrow \left[\mathrm{NN}_i(\text{inputs})\right]^2$
end
Compute the final early near-end signal
$\widehat{\mathbf{s}}_\mathrm{e}(n,f) \leftarrow$ (7)+(9)+(16)
Output:
$\widehat{\mathbf{s}}_\mathrm{e}(n,f)$

4 NN spectral model

In this section, we provide details for the derivation of inputs, targets and architecture of the NN 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 $\left[\sqrt{v_{s_e}(n,f)}\sqrt{v_{s_r}(n,f)}\sqrt{v_{z_r}(n,f)}\sqrt{v_{b_r}(n,f)}\right]$ as the targets for the NN. 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.

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 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.

At initialization, we set the three residual signals at the value of the original distortion signals:

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

$$\mathbf{z}_{\mathrm{r}}(n,f) \leftarrow \mathbf{y}(n,f), \tag{97}$$

$$\mathbf{b}_{\mathrm{r}}(n,f) \leftarrow \mathbf{b}(n,f). \tag{98}$$

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 steps 2 and 3 of Fig. 3 respectively. In order to update $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$, we apply the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ separately 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).

To obtain 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 replace NN-EM at step 4 of Fig. 3 by an *oracle* estimation. However, we do not use (95) on the target speech $\mathbf{s}_\mathrm{e}(n,f)$ and the latent signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$, as this equation does not integrate the MCSs $\mathbf{R}_{s_\mathrm{e}}(f)$, $\mathbf{R}_{s_\mathrm{r}}(f)$, $\mathbf{R}_{z_\mathrm{r}}(f)$ and $\mathbf{R}_{b_\mathrm{r}}(f)$. Instead, in the iterative procedure, we propose to use one iteration Duong et al.'s EM algorithm on each 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 $\mathbf{b}_\mathrm{r}(n,f)$ to jointly update the PSDs and the SCMs [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

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

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

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

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

and the SCMs as

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

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

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

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

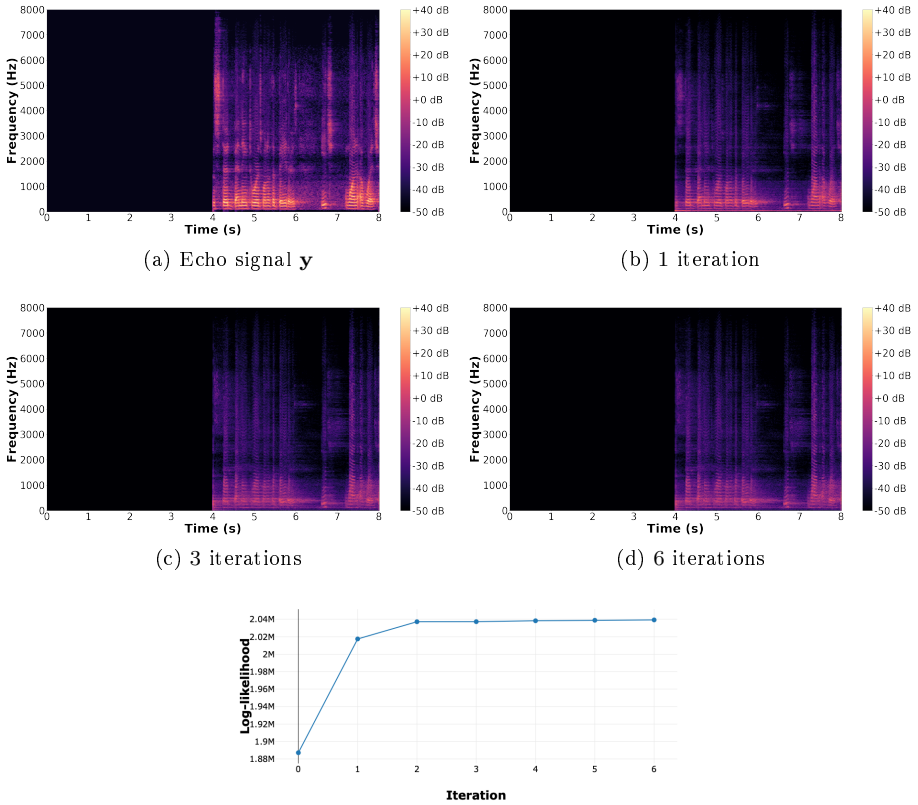
The SCMs are then normalized so as to encode only the spatial information of the sources as in (91)–(94).

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.

The detailed pseudo-code of the ground truth estimation procedure is provided in Alg. 2. 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.

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:
$\left[v_{s_{\mathrm{e}}}(n,f)\ v_{s_{\mathrm{r}}}(n,f)\ v_{z_{\mathrm{r}}}(n,f)\ v_{b_{\mathrm{r}}}(n,f)\right]$



(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_r} obtained by Alg. 2.

4.2 Inputs

We use magnitude spectra as inputs for NN_0 and NN_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 NN_0 and NN_i are summarized in Fig. 5. We concatenate these spectra to obtain the inputs. For NN_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 $|\widetilde{y}(n,f)|$, $|\widetilde{e}(n,f)|$, $|\widetilde{e}_\mathrm{l}(n,f)|$ and $|\widetilde{r}(n,f)|$ obtained from the corresponding multichannel signals after each linear filtering step $\widehat{\mathbf{y}}(n,f)$,

$\mathbf{e}(n,f)$, $\widehat{\mathbf{e}}_\mathrm{l}(n,f)$, $\mathbf{r}(n,f)$. 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 [12]. We refer to the above inputs as type-I inputs. The inputs of NN_0 are concatenated as

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

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

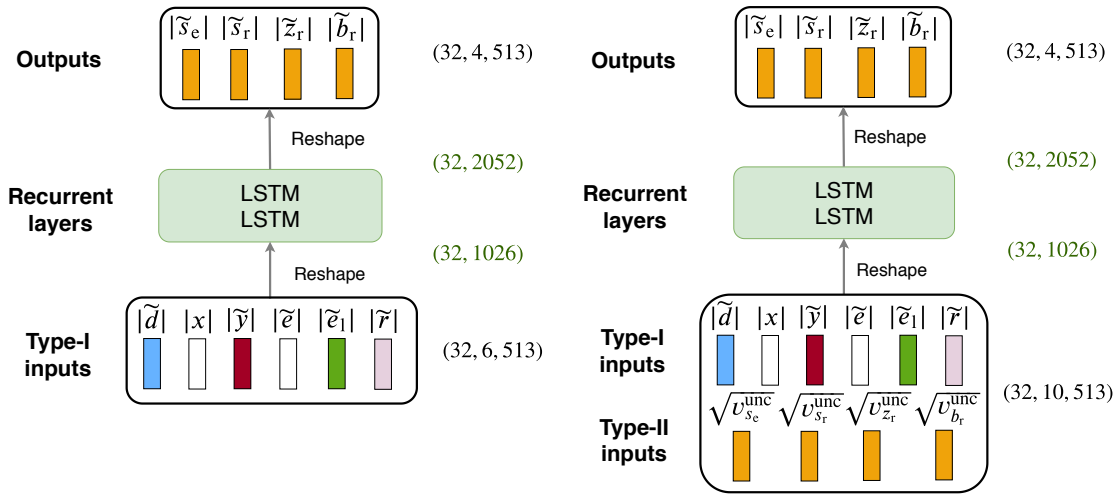
$$v_c^{\mathrm{unc}}(n,f) = \frac{1}{M}\mathrm{tr}\left(\mathbf{R}_c(f)^{-1}\widehat{\mathbf{R}}_c(n,f)\right). \tag{113}$$

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 NN_i are concatenated as

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

4.3 Architecture

The neural network follows a long-short-term-memory (LSTM) network architecture. We consider 2 LSTM layers (see Fig. 5). The number of inputs is $6F$ for NN_0 and $10F$ for NN_i. The number of outputs is $4F$. Other network architectures are not considered here as the performance comparison between different architectures is beyond the scope of this article.



(a) NN_0 (b) NN_i, $i \geq 1$

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

5 Variant with parallel echo cancellation and dereverberation

5.1 Model and expression of the likelihood

5.1.1 Model

We propose a variant of the joint approach where the echo cancellation and dereverberation filters are not applied one after another as in Fig. 2, but in parallel. The approach is illustrated in Fig. 6. In the first step, we apply the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in (**??**) and subtract the resulting echo estimate $\widehat{\mathbf{y}}(n,f)$ from $\mathbf{d}(n,f)$. In parallel, we apply the dereverberation filter $\underline{\mathbf{G}}(f)$ on the mixture signal $\mathbf{d}(n,f)$ and subtract the resulting late reverberation estimate $\widehat{\mathbf{d}}_{\mathrm{l}}(n,f)$ from $\mathbf{d}(n,f)$. The resulting signal $\mathbf{r}(n,f)$ after echo cancellation and dereverberation is then

$$\mathbf{e}(n,f) = \mathbf{d}(n,f) - \underbrace{\sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f)}_{=\widehat{\mathbf{y}}(\mathbf{n},\mathbf{f})} - \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{d}(n-l,f)}_{=\widehat{\mathbf{d}}_{\mathrm{l}}(n,f)}. \tag{115}$$

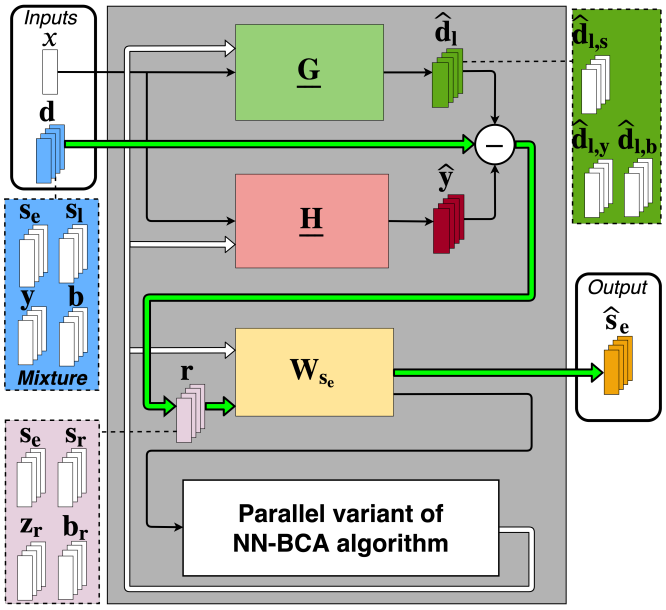


Figure 6: Parallel variant of the proposed approach for joint reduction of echo, reverberation and noise where the echo cancellation and dereverberation filters are applied in parallel. Arrows and lines have the same meaning as in Fig. 2.

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 $n<0$. Similarly to (10), the signal $\mathbf{r}(n,f)$ comprises the target early near-end signal component $s_{\mathrm{e}}(n,f)$, 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 $\mathbf{b}_{\mathrm{r}}(n,f)$:

$$\mathbf{r}(n,f)=\mathbf{s}_{\mathrm{e}}(n,f)+\mathbf{s}_{\mathrm{r}}(n,f)+\mathbf{z}_{\mathrm{r}}(n,f)+\mathbf{b}_{\mathrm{r}}(n,f). \tag{116}$$

The three residual signals are expressed differently than (11)–(13):

$$\mathbf{s}_{\mathrm{r}}(n,f)=\mathbf{s}_{\mathrm{l}}(n,f)-\widehat{\mathbf{d}}_{\mathrm{l},s}(n,f), \tag{117}$$

$$\mathbf{z}_{\mathrm{r}}(n,f)=\mathbf{y}(n,f)-\widehat{\mathbf{y}}(n,f)-\widehat{\mathbf{d}}_{\mathrm{l},y}(n,f), \tag{118}$$

$$\mathbf{b}_{\mathrm{r}}(n,f)=\mathbf{b}(n,f)-\widehat{\mathbf{d}}_{\mathrm{l},b}(n,f), \tag{119}$$

where the signals $\widehat{\mathbf{d}}_{\mathrm{l},s}(n,f)$, $\widehat{\mathbf{d}}_{\mathrm{l},y}(n,f)$ and $\widehat{\mathbf{d}}_{\mathrm{l},b}(n,f)$ are the latent components of $\widehat{\mathbf{d}}_{\mathrm{l}}(n,f)$ resulting from (115)

$$\widehat{\mathbf{d}}_{\mathrm{l}}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\Big(\mathbf{s}(n-l,f)+\mathbf{y}(n-l,f)+\mathbf{b}(n-l,f)\Big) \tag{120}$$

$$= \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{s}(n-l,f)}_{=\widehat{\mathbf{d}}_{\mathrm{l},s}(n,f)} + \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{y}(n-l,f)}_{=\widehat{\mathbf{d}}_{\mathrm{l},y}(n,f)} + \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{b}(n-l,f)}_{=\widehat{\mathbf{d}}_{\mathrm{l},b}(n,f)}. \tag{121}$$

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

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f) = \mathbf{W}_{s_{\mathrm{e}}}(n,f)\mathbf{r}(n,f). \tag{122}$$

Similarly to the originally-proposed approach, 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 similarly to (17)–(20):

$$\mathbf{s}_{\mathrm{e}}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_{s_{\mathrm{e}}}(n,f)\mathbf{R}_{s_{\mathrm{e}}}(f)\Big), \tag{123}$$

$$\mathbf{s}_{\mathrm{r}}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_{s_{\mathrm{r}}}(n,f)\mathbf{R}_{s_{\mathrm{r}}}(f)\Big), \tag{124}$$

$$\mathbf{z}_{\mathrm{r}}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_{z_{\mathrm{r}}}(n,f)\mathbf{R}_{z_{\mathrm{r}}}(f)\Big), \tag{125}$$

$$\mathbf{b}_{\mathrm{r}}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_{b_{\mathrm{r}}}(n,f)\mathbf{R}_{b_{\mathrm{r}}}(f)\Big). \tag{126}$$

The multichannel Wiener filter for the target $\mathbf{s}_{\mathrm{e}}(n,f)$ is thus formulated as

$$\mathbf{W}_{s_{\mathrm{e}}}(n,f) = v_{s_{\mathrm{e}}}(n,f)\mathbf{R}_{s_{\mathrm{e}}}(f)\left(\sum_{c'\in\mathcal{C}'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{127}$$

where $\mathcal{C}' = \{\mathbf{s}_\mathrm{e}, \mathbf{s}_\mathrm{r}, \mathbf{z}_\mathrm{r}, \mathbf{b}_\mathrm{r}\}$ denotes all four sources in signal $\mathbf{r}(n,f)$. 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

$$\mathbf{W}_{s_\mathrm{r}}(n,f) = v_{s_\mathrm{r}}(n,f)\mathbf{R}_{s_\mathrm{r}}(f)\left(\sum_{c'\in\mathcal{C}'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{128}$$

$$\mathbf{W}_{z_\mathrm{r}}(n,f) = v_{z_\mathrm{r}}(n,f)\mathbf{R}_{z_\mathrm{r}}(f)\left(\sum_{c'\in\mathcal{C}'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{129}$$

$$\mathbf{W}_{b_\mathrm{r}}(n,f) = v_{b_\mathrm{r}}(n,f)\mathbf{R}_{b_\mathrm{r}}(f)\left(\sum_{c'\in\mathcal{C}'} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{130}$$

respectively.

5.1.2 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

$$\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f) \sim \mathcal{N}_\mathbb{C}\Big(\mathbf{d}(n,f);\boldsymbol{\mu}'_\mathbf{d}(n,f),\mathbf{R}'_\mathbf{dd}(n,f)\Big), \tag{131}$$

where

$$\boldsymbol{\mu}'_\mathbf{d}(n,f) = \sum_{k=0}^{K-1}\mathbf{h}(k,f)x(n-k,f) + \sum_{l=\Delta}^{\Delta+L-1}\mathbf{G}(l,f)\mathbf{d}(n-l,f) \tag{132}$$

$$\mathbf{R}'_\mathbf{dd}(n,f) = \sum_{c'\in\mathcal{C}'} v_{c'}(n,f)\mathbf{R}_{c'}(n,f). \tag{133}$$

The parameters to be estimated are denoted by

$$\Theta_H = \{\underline{\mathbf{H}}(f)\}_f \tag{134}$$

$$\Theta_G = \{\underline{\mathbf{G}}(f)\}_f, \tag{135}$$

$$\Theta_c = \Big\{v_{s_\mathrm{e}}(n,f), \mathbf{R}_{s_\mathrm{e}}(f), v_{s_\mathrm{r}}(n,f), \mathbf{R}_{s_\mathrm{r}}(f), v_{z_\mathrm{r}}(n,f), \mathbf{R}_{z_\mathrm{r}}(f), v_{b_\mathrm{r}}(n,f), \mathbf{R}_{b_\mathrm{r}}(f)\Big\}_{n,f}. \tag{136}$$

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

$$\mathcal{L}\Big(\mathcal{O}; \Theta_H, \Theta_G, \Theta_c\Big) = \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log p\Big(\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f)\Big), \tag{137}$$

$$= \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log \mathcal{N}_{\mathbb{C}}\Big(\mathbf{d}(n,f); \boldsymbol{\mu}'_{\mathbf{d}}(n,f), \mathbf{R}'_{\mathbf{dd}}(n,f)\Big) \tag{138}$$

$$\equiv \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log |\mathbf{R}'_{\mathbf{dd}}(n,f)|^{-1} - \Big(\mathbf{d}(n,f) - \boldsymbol{\mu}'_{\mathbf{d}}(n,f)\Big)^H \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\Big(\mathbf{d}(n,f) - \boldsymbol{\mu}'_{\mathbf{d}}(n,f)\Big). \tag{139}$$

5.2 Vectorized computation of echo cancellation and dereveberation

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, we reformulate the term $\widehat{\mathbf{y}}(n,f) + \mathbf{d}_{\mathrm{l}}(n,f) = \sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f) + \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{d}(n-l,f)$ as one filtering process:

$$\widehat{\mathbf{y}}(n,f) + \mathbf{d}_{\mathrm{l}}(n,f) = \underline{\boldsymbol{\chi}}(n,f)\underline{\boldsymbol{\phi}}(f), \tag{140}$$

where $\underline{\boldsymbol{\phi}}(f) \in \mathbb{C}^{(MK+M^2L)\times 1}$ is a vectorized version of the concatenated filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ as

$$\underline{\boldsymbol{\phi}}(f) = \begin{bmatrix} \mathbf{h}(0,f) \\ \vdots \\ \mathbf{h}(K-1,f) \\ \mathbf{g}_1(\Delta,f) \\ \vdots \\ \mathbf{g}_M(\Delta,f) \\ \vdots \\ \vdots \\ \mathbf{g}_1(\Delta+L-1,f) \\ \vdots \\ \mathbf{g}_M(\Delta+L-1,f) \end{bmatrix}, \tag{141}$$

$\underline{\boldsymbol{\chi}}(n,f) \in \mathbb{C}^{M\times(MK+M^2L)}$ is the concatenation of the K taps $\mathbf{X}(n-k,f) \in \mathbb{C}^{M\times M}$ and the L taps $\mathbf{D}(n-l,f) \in \mathbb{C}^{M\times M^2}$ as

$$\underline{\boldsymbol{\chi}}(n,f) = \Big[\mathbf{X}(n,f)\dots\mathbf{X}(n-K+1,f)\mathbf{D}(n-\Delta,f)\dots\mathbf{D}(n-\Delta-L+1,f)\Big], \tag{142}$$

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

$$\mathbf{X}(n-k,f) = x(n-k,f)\mathbf{I}_M, \tag{143}$$

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

$$\mathbf{D}(n-l,f) = \mathbf{I}_M \otimes \mathbf{d}(n-l,f)^T. \tag{144}$$

The resulting ML optimization problem has no closed-form solution, hence we need to estimate the parameters via a variant of the NN-supported BCA algorithm.

5.3 Iterative optimization algorithm

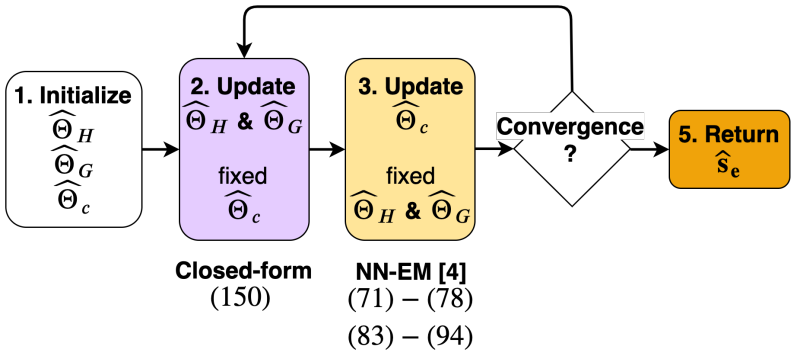


Figure 7: Flowchart of the variant of NN-supported BCA algorithm for likelihood optimization.

We propose a BCA algorithm for likelihood optimization. Each iteration i comprises the following two maximization steps:

$$\widehat{\Theta}_H, \widehat{\Theta}_G \leftarrow \underset{\Theta_H}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \Theta_H, \Theta_G, \widehat{\Theta}_c\right), \tag{145}$$

$$\widehat{\Theta}_c \leftarrow \underset{\Theta_c}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \widehat{\Theta}_G, \Theta_c\right). \tag{146}$$

The solution of (145) is closed-form. Note that the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are estimated as a unique filter $\underline{\boldsymbol{\phi}}(f)$ due to (140). As there is no closed-form solution for (146), we propose to use Nugraha et al.'s NN-EM algorithm [4]. The overall flowchart of the proposed algorithm is shown in Fig. 7. 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 (116) [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 (145)–(146) of our proposed algorithm at iteration i.

5.3.1 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 $v_{s_\mathrm{e}}(n,f)$, $v_{s_\mathrm{r}}(n,f)$, $v_{z_\mathrm{r}}(n,f)$ and $v_{b_\mathrm{r}}(n,f)$ of the target and residual signals are jointly initialized using a pretrained NN denoted as NN_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 NN_0 are described in Section

5.4 below.

5.3.2 Echo cancellation filter parameters Θ_H and dereverberation parameters Θ_G

As a reminder, the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are estimated as a unique filter $\underline{\boldsymbol{\phi}}(f)$ due to (140). The partial derivative of the log-likelihood with respect to $\underline{\boldsymbol{\phi}}(f)$ can be computed as

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\boldsymbol{\phi}}(f)} = -\frac{\partial}{\partial \underline{\boldsymbol{\phi}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{d}}_{\mathrm{l}}(n,f)\Bigg)^H \\ \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\Bigg(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \widehat{\mathbf{d}}_{\mathrm{l}}(n,f)\Bigg).\end{aligned} \tag{147}$$

Replacing (140) in (147):

$$\begin{aligned}\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\right)}{\partial \underline{\boldsymbol{\phi}}(f)} &= -\frac{\partial}{\partial \underline{\boldsymbol{\phi}}(f)} \sum_{n=0}^{N-1} \Bigg(\mathbf{d}(n,f) - \underline{\boldsymbol{\chi}}(n,f)\underline{\boldsymbol{\phi}}(f)\Bigg)^H \\ &\quad \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\Bigg(\mathbf{d}(n,f) - \underline{\boldsymbol{\chi}}(n,f)\underline{\boldsymbol{\phi}}(f)\Bigg) \qquad (148)\\ &= \sum_{n=0}^{N-1} 2\underline{\boldsymbol{\chi}}(n,f)^H \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\underline{\boldsymbol{\chi}}(n,f)\underline{\boldsymbol{\phi}}(f) \\ &\quad - 2\underline{\boldsymbol{\chi}}(n,f)^H \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\mathbf{d}(n,f). \qquad (149)\end{aligned}$$

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

$$\underline{\boldsymbol{\phi}}(f) = \boldsymbol{\Psi}(f)^{-1}\boldsymbol{\psi}(f), \tag{150}$$

where

$$\boldsymbol{\Psi}(f) = \sum_{n=0}^{N-1} \underline{\boldsymbol{\chi}}(n,f)^H \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\underline{\boldsymbol{\chi}}(n,f) \tag{151}$$

$$\boldsymbol{\psi}(f) = \sum_{n=0}^{N-1} \underline{\boldsymbol{\chi}}(n,f)^H \mathbf{R}'_{\mathbf{dd}}(n,f)^{-1}\mathbf{d}(n,f). \tag{152}$$

Note that the matrix $\underline{\boldsymbol{\Psi}}(n,f)$ is a sum of rank-M terms, thus requires at least $ML+K$ terms in order to be invertible. Updating the filter $\underline{\boldsymbol{\phi}}(f)$ enables to jointly update the echo cancellation

filter $\underline{\mathbf{H}}(f)$ and the dereverberation filter $\underline{\mathbf{G}}(f)$. The echo cancellation filter $\underline{\mathbf{H}}(f)$ and the dereverberation filter $\underline{\mathbf{G}}(f)$ are both influenced by each other through the terms $\underline{\boldsymbol{\phi}}(f)$.

5.3.3 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 are estimated with the NN-EM algorithm described in Section 3.4.

5.3.4 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 estimates of the PSDs $v_c(n, f)$ and the SCMs $\mathbf{R}_c(f)$. We can perform one more iteration of the parallel variant of the NN-BCA algorithm to derive the final filters $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_e}(n, f)$. Ultimately, we obtain the target estimate $\widehat{\mathbf{s}}_{\mathrm{e}}(n, f)$ using (115), and (122). The detailed pseudo-code of the algorithm is provided in Alg. 3.

Algorithm 3: Proposed variant of NN-supported BCA algorithm for joint reduction of echo, reverberation and noise.

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

Initialize:
- Initialize the linear filters
  - $\underline{\mathbf{h}}(f) \leftarrow \underline{\mathbf{h}}_0(f)$ (chosen by the user, e.g. [11])
  - $\underline{\mathbf{g}}(f) \leftarrow \underline{\mathbf{g}}_0(f)$ (chosen by the user, e.g. WPE)
- Initialize the SCMs
  - $\left[\mathbf{R}_{s_\mathrm{e}}(f)\ \mathbf{R}_{s_\mathrm{r}}(f)\ \mathbf{R}_{z_\mathrm{r}}(f)\ \mathbf{R}_{b_\mathrm{r}}(f)\right] \leftarrow \left[\mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\right]$
- Initialize the NN inputs
  - inputs $\leftarrow$ (112)
- Initialize the PSDs
  - $\left[v_{s_\mathrm{e}}(n,f)\ v_{s_\mathrm{r}}(n,f)\ v_{z_\mathrm{r}}(n,f)\ v_{b_\mathrm{r}}(n,f)\right] \leftarrow \left[\mathrm{NN}_0(\text{inputs})\right]^2$

for *each iteration i of I* **do**
- Update both the echo cancellation and dereverberation filters
  - $\underline{\boldsymbol{\phi}}(f) \leftarrow$ (150)
- Update signal $\mathbf{r}(n,f)$
  - $\mathbf{r}(n,f) \leftarrow$ (115)
- Update the SCMs
- **for** *each spatial update j of J* **do**
  - **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 multichannel Wiener filter
      - $\mathbf{W}_c(n,f) \leftarrow$ (21) or (22) or (23) or (24)
    - Update the source estimation
      - $\widehat{\mathbf{c}}(n,f) \leftarrow$ (16) 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 NN inputs
  - inputs $\leftarrow$ (114)
- Update the PSDs
  - $\left[v_{s_\mathrm{e}}(n,f)\ v_{s_\mathrm{r}}(n,f)\ v_{z_\mathrm{r}}(n,f)\ v_{b_\mathrm{r}}(n,f)\right] \leftarrow \left[\mathrm{NN}_i(\text{inputs})\right]^2$

end

Compute the final early near-end signal
- $\widehat{\mathbf{s}}_\mathrm{e}(n,f) \leftarrow$ (7)+(9)+(16)

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

5.4 NN spectral model

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

5.4.1 Targets

In order to obtain 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 use the same procedure as in Section 4.1. The only difference lies in the update of the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$. Instead of using (60) and (67), respectively, we update the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ using (150).

The detailed pseudo-code of the ground truth estimation procedure is provided in Alg. 4. 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. The signals $\mathbf{s}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ did not change after 1 iteration.

5.4.2 Inputs

We use the same inputs as in Section 4.2. However, the type-I input for $|\widetilde{e}_{\mathrm{l}}(n,f)|$ is replaced by $|\widetilde{d}_{\mathrm{l}}(n,f)|$ obtained similarly to (111) from the corresponding multichannel signal $\widehat{\mathbf{d}}_{\mathrm{l}}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{d}(n-l,f)$ (see Fig. 6).

Algorithm 4: 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 both the echo cancellation and dereverberation filters
$\underline{\phi}(f) \leftarrow$ (150)
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:
$\left[v_{s_{\mathrm{e}}}(n,f)\ v_{s_{\mathrm{r}}}(n,f)\ v_{z_{\mathrm{r}}}(n,f)\ v_{b_{\mathrm{r}}}(n,f)\right]$

6 Variant with only NN-supported echo cancellation

6.1 Model and expression of the likelihood

6.1.1 Model

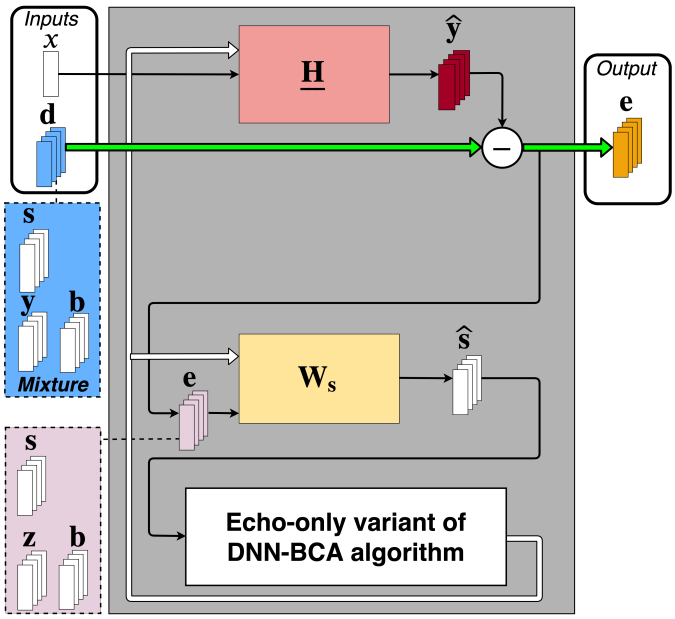


Figure 8: Echo-only variant of the proposed approach for joint reduction of echo, reverberation and noise where only the echo cancellation. Arrows and lines have the same meaning as in Fig. 2.

Here, the goal would be to recover the near-end speech $\mathbf{s}(n,f)$ from the mixture $\mathbf{d}(n,f)$. We propose a variant of the joint approach where only the echo cancellation filter and the post-filter are applied. Although the ultimate goal is actually to recover the signal $\mathbf{e}(n,f)$ after echo cancellation, the post-filter is required in order to obtain an echo cancellation filter $\underline{\mathbf{H}}(f)$ comparable to our initially-proposed approach. Therefore, throughout this section, we describe the building blocks of the variant as though we aimed at recovering the near-end speech $\mathbf{s}(n,f)$.

The approach is illustrated in Fig. 8. In the first step, we apply the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in (7) and subtract the resulting echo estimate $\widehat{\mathbf{y}}(n,f)$ from $\mathbf{d}(n,f)$. The resulting signal

$\mathbf{e}(n,f)$ after echo cancellation is then

$$\mathbf{e}(n,f) = \mathbf{d}(n,f) - \underbrace{\sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f)}_{=\widehat{\mathbf{y}}(\mathbf{n},\mathbf{f})}. \tag{153}$$

Since the linear filter $\underline{\mathbf{H}}(f)$ is applied on the previous frames of the signal $x(n,f)$, we make the assumption that the observed signal $x(n,f)$ is equal to zero for $n<0$. Similarly to (8), the signal $\mathbf{e}(n,f)$ comprises the near-end signal component $s(n,f)$, the residual echo $\mathbf{z}(n,f)$, and the noise $\mathbf{b}(n,f)$:

$$\mathbf{e}(n,f) = \mathbf{s}(n,f) + \mathbf{z}(n,f) + \mathbf{b}(n,f). \tag{154}$$

The residual echo is expressed as:

$$\mathbf{z}(n,f) = \mathbf{y}(n,f) - \widehat{\mathbf{y}}(n,f). \tag{155}$$

To recover $\mathbf{s}(n,f)$ from $\mathbf{e}(n,f)$, we apply a multichannel Wiener postfilter $\mathbf{W}_s(n,f)$:

$$\widehat{\mathbf{s}}(n,f) = \mathbf{W}_s(n,f)\mathbf{e}(n,f). \tag{156}$$

Similarly to the originally-proposed approach, we estimate $\underline{\mathbf{H}}(f)$ and $\mathbf{W}_s(n,f)$ in the maximum likelihood (ML) sense by modeling the target $\mathbf{s}(n,f)$ and the two distortion signals $\mathbf{z}(n,f)$ and $\mathbf{b}(n,f)$ with a multichannel local Gaussian framework. We thus consider each of these three signals as *sources* to be separated and model them similarly to (17)–(20):

$$\mathbf{s}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_s(n,f)\mathbf{R}_s(f)\Big), \tag{157}$$

$$\mathbf{z}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_z(n,f)\mathbf{R}_z(f)\Big), \tag{158}$$

$$\mathbf{b}(n,f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0}, v_b(n,f)\mathbf{R}_b(f)\Big). \tag{159}$$

The multichannel Wiener filter for the target $\mathbf{s}(n,f)$ is thus formulated as

$$\mathbf{W}_s(n,f) = v_s(n,f)\mathbf{R}_s(f)\left(\sum_{c'\in\mathcal{C}''} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{160}$$

where $\mathcal{C}'' = \{\mathbf{s}, \mathbf{z}, \mathbf{b}\}$ denotes all three sources in signal $\mathbf{e}(n, f)$. Similarly, the multichannel Wiener filters for the residual signals $\mathbf{z}(n, f)$ and $\mathbf{b}(n, f)$ are formulated as

$$\mathbf{W}_z(n, f) = v_z(n, f)\mathbf{R}_z(f) \left(\sum_{c' \in \mathcal{C}''} v_{c'}(n, f)\mathbf{R}_{c'}(f) \right)^{-1}, \tag{161}$$

$$\mathbf{W}_b(n, f) = v_b(n, f)\mathbf{R}_b(f) \left(\sum_{c' \in \mathcal{C}''} v_{c'}(n, f)\mathbf{R}_{c'}(f) \right)^{-1}, \tag{162}$$

respectively.

6.1.2 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

$$\mathbf{d}(n, f) \Big| \mathbf{d}(n-1, f), \ldots, \mathbf{d}(0, f), x(n, f), \ldots, x(0, f) \sim \mathcal{N}_{\mathbb{C}} \Big(\mathbf{d}(n, f); \boldsymbol{\mu}''_{\mathbf{d}}(n, f), \mathbf{R}''_{\mathbf{dd}}(n, f) \Big), \tag{163}$$

where

$$\boldsymbol{\mu}''_{\mathbf{d}}(n, f) = \sum_{k=0}^{K-1} \mathbf{h}(k, f) x(n-k, f) \tag{164}$$

$$\mathbf{R}''_{\mathbf{dd}}(n, f) = \sum_{c' \in \mathcal{C}'} v_{c'}(n, f)\mathbf{R}_{c'}(n, f). \tag{165}$$

The parameters to be estimated are denoted by

$$\Theta_H = \{\underline{\mathbf{H}}(f)\}_f \tag{166}$$

$$\Theta_c = \Big\{ v_s(n, f), \mathbf{R}_s(f), v_z(n, f), \mathbf{R}_z(f), v_b(n, f), \mathbf{R}_b(f) \Big\}_{n,f}. \tag{167}$$

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

$$\mathcal{L}\Big(\mathcal{O};\Theta_H,\Theta_c\Big) = \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log p\Big(\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f)\Big), \tag{168}$$

$$= \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log \mathcal{N}_{\mathbb{C}}\Big(\mathbf{d}(n,f);\boldsymbol{\mu}''_{\mathbf{d}}(n,f),\mathbf{R}''_{\mathbf{dd}}(n,f)\Big) \tag{169}$$

$$\begin{aligned} \equiv \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} & \log|\mathbf{R}''_{\mathbf{dd}}(n,f)|^{-1} \\ & - \Big(\mathbf{d}(n,f)-\boldsymbol{\mu}''_{\mathbf{d}}(n,f)\Big)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1}\Big(\mathbf{d}(n,f)-\boldsymbol{\mu}''_{\mathbf{d}}(n,f)\Big). \end{aligned} \tag{170}$$

6.2 Vectorized computation of echo cancellation

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, we reformulate the term $\widehat{\mathbf{y}}(n,f) = \sum_{k=0}^{K-1}\mathbf{h}(k,f)x(n-k,f)$ as in (34)

$$\widehat{\mathbf{y}}(n,f) = \underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f), \tag{171}$$

where $\underline{\mathbf{h}}(f) \in \mathbb{C}^{MK\times 1}$ is a vectorized version of the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in (35), and $\underline{\mathbf{X}}(n,f) \in \mathbb{C}^{M\times MK}$ is the concatenation of the K taps $\mathbf{X}(n-k,f) \in \mathbb{C}^{M\times M}$ as in (36).

The resulting ML optimization problem has no closed-form solution, hence we need to estimate the parameters via a variant of the NN-supported BCA algorithm.

6.3 Iterative optimization algorithm

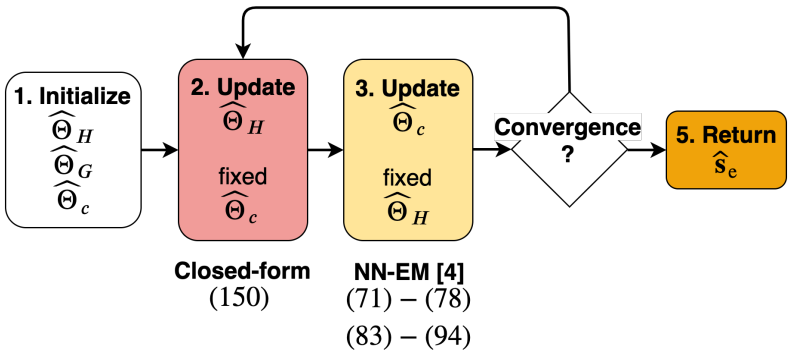


Figure 9: Flowchart of the variant of NN-supported BCA algorithm for likelihood optimization.

We propose a BCA algorithm for likelihood optimization. Each iteration i comprises the following two maximization steps:

$$\widehat{\Theta}_H \leftarrow \underset{\Theta_H}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \Theta_H, \widehat{\Theta}_c\right), \tag{172}$$

$$\widehat{\Theta}_c \leftarrow \underset{\Theta_c}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \Theta_c\right). \tag{173}$$

The solution of (172) is closed-form. As there is no closed-form solution for (173), we propose to use Nugraha et al.'s NN-EM algorithm [4]. The overall flowchart of the proposed algorithm is shown in Fig. 9. Note that it is also possible to optimize the parameters Θ_H and Θ_c with the EM algorithm by adding a noise term to (154) [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 (172)–(173) of our proposed algorithm at iteration i.

6.3.1 Initialization

We initialize the linear filters $\underline{\mathbf{H}}(f)$ to $\underline{\mathbf{H}}_0(f)$. The PSDs $v_s(n,f)$, $v_z(n,f)$ and $v_b(n,f)$ of the target and distortion signals are jointly initialized using a pretrained NN denoted as NN_0 and the SCMs $\mathbf{R}_s(f)$, $\mathbf{R}_z(f)$ and $\mathbf{R}_b(f)$ as the identity matrix $\mathbf{I}_M$. The inputs, the targets and the architecture of NN_0 are described in Section 6.4 below.

6.3.2 Echo cancellation filter parameters Θ_H

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

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f)\right)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1} \left(\mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f)\right). \tag{174}$$

Replacing (171) in (174):

$$\frac{\partial \mathcal{L}\left(\mathcal{O};\Theta_H,\Theta_c\right)}{\partial \underline{\mathbf{h}}(f)} = -\frac{\partial}{\partial \underline{\mathbf{h}}(f)} \sum_{n=0}^{N-1} \left(\mathbf{d}(n,f) - \underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f)\right)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1} \left(\mathbf{d}(n,f) - \underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f)\right) \tag{175}$$

$$= \sum_{n=0}^{N-1} 2\underline{\mathbf{X}}(n,f)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{X}}(n,f)\underline{\mathbf{h}}(f) - 2\underline{\mathbf{X}}(n,f)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1}\mathbf{d}(n,f). \tag{176}$$

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

$$\underline{\mathbf{h}}(f) = \mathbf{P}'^{-1}\mathbf{p}'(f), \tag{177}$$

where

$$\mathbf{P}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}(n,f)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{X}}(n,f) \tag{178}$$

$$\mathbf{p}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}(n,f)^H \mathbf{R}''_{\mathbf{dd}}(n,f)^{-1} \mathbf{d}(n,f). \tag{179}$$

Note that the matrix $\underline{\mathbf{P}}(n,f)$ is a sum of rank-M terms, thus requires at least K terms in order to be invertible.

6.3.3 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 $\underline{\mathbf{H}}(f)$, the signal $\mathbf{e}(n,f)$ is conditionally distributed as

$$\mathbf{e}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f),\underline{\mathbf{H}}(f) \sim \mathcal{N}_{\mathbb{C}}\Big(\mathbf{0},\mathbf{R}''_{\mathbf{dd}}(n,f)\Big). \tag{180}$$

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 $\underline{\mathbf{H}}(f)$, we propose to use one iteration of Nugraha et al.'s NN-EM algorithm to update the PSDs and the SCMs of the target and distortion signals $\mathbf{s}(n,f)$, $\mathbf{z}(n,f)$ and $\mathbf{b}(n,f)$ [4]. In the E-step, the target and residual signals are estimated as

$$\widehat{\mathbf{s}}(n,f) = \mathbf{W}_s(n,f)\mathbf{e}(n,f), \tag{181}$$

$$\widehat{\mathbf{z}}(n,f) = \mathbf{W}_z(n,f)\mathbf{e}(n,f), \tag{182}$$

$$\widehat{\mathbf{b}}(n,f) = \mathbf{W}_b(n,f)\mathbf{e}(n,f), \tag{183}$$

and their second-order posterior moments as

$$\widehat{\mathbf{R}}_s(n,f) = \widehat{\mathbf{s}}(n,f)\widehat{\mathbf{s}}(n,f)^H + \Big(\mathbf{I}-\mathbf{W}_s(n,f)\Big)v_s(n,f)\mathbf{R}_s(f), \tag{184}$$

$$\widehat{\mathbf{R}}_z(n,f) = \widehat{\mathbf{z}}(n,f)\widehat{\mathbf{z}}(n,f)^H + \Big(\mathbf{I}-\mathbf{W}_z(n,f)\Big)v_z(n,f)\mathbf{R}_z(f), \tag{185}$$

$$\widehat{\mathbf{R}}_b(n,f) = \widehat{\mathbf{b}}(n,f)\widehat{\mathbf{b}}(n,f)^H + \Big(\mathbf{I}-\mathbf{W}_b(n,f)\Big)v_b(n,f)\mathbf{R}_b(f). \tag{186}$$

In the M-step, we update the SCMs $\mathbf{R}_s(f)$, $\mathbf{R}_z(f)$ and $\mathbf{R}_b(f)$ using the weighted form [8]:

$$\mathbf{R}_s(f) = \left(\sum_{n=0}^{N-1} w_s(n,f)\right)^{-1} \sum_{n=0}^{N-1} \frac{w_s(n,f)}{v_s(n,f)} \widehat{\mathbf{R}}_s(n,f), \tag{187}$$

$$\mathbf{R}_z(f) = \left(\sum_{n=0}^{N-1} w_z(n,f)\right)^{-1} \sum_{n=0}^{N-1} \frac{w_z(n,f)}{v_z(n,f)} \widehat{\mathbf{R}}_z(n,f), \tag{188}$$

$$\mathbf{R}_b(f) = \left(\sum_{n=0}^{N-1} w_b(n,f)\right)^{-1} \sum_{n=0}^{N-1} \frac{w_b(n,f)}{v_b(n,f)} \widehat{\mathbf{R}}_b(n,f), \tag{189}$$

where $w_s(n,f)$, $w_z(n,f)$ and $w_b(n,f)$ denote the weight of the target and distortion signals $\mathbf{s}(n,f)$, $\mathbf{z}(n,f)$ and $\mathbf{b}(n,f)$, which we define as follows [8, 9]

$$w_s(n,f) = v_s(n,f), \tag{190}$$

$$w_z(n,f) = v_z(n,f), \tag{191}$$

$$w_b(n,f) = v_b(n,f). \tag{192}$$

Experience shows that this weighting trick mitigates inaccurate estimates in certain time-frequency bins and increases the importance of the bins for which $v_s(n,f)$, $v_z(n,f)$ and $v_b(n,f)$ are large. As the PSDs are constrained, we also need to constrain $\mathbf{R}_s(f)$, $\mathbf{R}_z(f)$ and $\mathbf{R}_b(f)$ so as to encode only the spatial information of the sources. We modify (187)–(189) by normalizing $\mathbf{R}_s(f)$, $\mathbf{R}_z(f)$ and $\mathbf{R}_b(f)$ after each update [8]:

$$\mathbf{R}_s(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_s(f)\right)} \mathbf{R}_s(f), \tag{193}$$

$$\mathbf{R}_z(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_z(f)\right)} \mathbf{R}_z(f), \tag{194}$$

$$\mathbf{R}_b(f) \leftarrow \frac{M}{\mathrm{tr}\left(\mathbf{R}_b(f)\right)} \mathbf{R}_b(f). \tag{195}$$

The PSDs $v_s(n,f)$, $v_z(n,f)$ and $v_b(n,f)$ are jointly updated using a pretrained NN denoted as NN_i, with $i \geq 1$ the iteration index. The inputs, the targets and the architecture of NN_i are described in Section 6.4 below.

6.3.4 Estimation of the final near-end speech $\mathbf{s}(n, f)$

Once the proposed iterative optimization algorithm has converged after I iterations, we have estimates of the PSDs $v_c(n, f)$ and the SCMs $\mathbf{R}_c(f)$ We can perform one more iteration of the echo-only variant of the NN-BCA algorithm to derive the final filters $\underline{\mathbf{H}}(f)$ and $\mathbf{W}_s(n, f)$. Ultimately, we obtain the target estimate $\widehat{\mathbf{s}}(n, f)$ using (153) and (156). The detailed pseudo-code of the algorithm is provided in Alg. 5.

Algorithm 5: Proposed variant of NN-BCA algorithm for echo reduction.

Input:
$\mathbf{d}(n,f), x(n,f)$
Pretrained $\mathrm{NN}_0, \mathrm{NN}_1, \ldots, \mathrm{NN}_I$
Initialize:
Initialize the echo cancellation filter
$\underline{\mathbf{h}}(f) \leftarrow \underline{\mathbf{h}}_0(f)$ (chosen by the user, e.g. [11])
Initialize the SCMs
$\left[\mathbf{R}_s(f)\ \mathbf{R}_z(f)\ \mathbf{R}_b(f)\right] \leftarrow \left[\mathbf{I}_M\ \mathbf{I}_M\ \mathbf{I}_M\right]$
Initialize the NN inputs
inputs $\leftarrow$ (206)
Initialize the PSDs
$\left[v_s(n,f)\ v_z(n,f)\ v_b(n,f)\right] \leftarrow \left[\mathrm{NN}_0(\text{inputs})\right]^2$
for *each iteration* i *of* I **do**
  Update both the echo cancellation filter
  $\underline{\mathbf{h}}(f) \leftarrow$ (177)
  Update signal $\mathbf{e}(n,f)$
  $\mathbf{e}(n,f) \leftarrow$ (153)
  Update the SCMs
  for *each spatial update* j *of* J **do**
    for *each source* $\mathbf{c}$ *of* $[\mathbf{s},\mathbf{z},\mathbf{b}]$ **do**
      Update the multichannel Wiener filter
      $\mathbf{W}_c(n,f) \leftarrow$ (160) or (161) or (162)
      Update the source estimation
      $\widehat{\mathbf{c}}(n,f) \leftarrow$ (156) or (182) or (183)
      Update the posterior statistics
      $\widehat{\mathbf{R}}_c(n,f) \leftarrow$ (184) or (185) or (186)
      Update the source SCM
      $\mathbf{R}_c(f) \leftarrow$ (187)+(190)+(193) or (188)+(191)+(194) or (189)+(192)+(195)
    end
  end
  Update the NN inputs
  inputs $\leftarrow$ (207)
  Update the PSDs
  $\left[v_s(n,f)\ v_z(n,f)\ v_b(n,f)\right] \leftarrow \left[\mathrm{NN}_i(\text{inputs})\right]^2$
end
Compute the final signal (n,f) and near-end signal
$\mathbf{e}(n,f) \leftarrow$ (153)
$\widehat{\mathbf{s}}(n,f) \leftarrow$ (156)
Output:
$\widehat{\mathbf{s}}(n,f)$

6.4 NN spectral model

6.4.1 Targets

In order to obtain the ground truth PSDs $v_s(n,f)$, $v_z(n,f)$ and $v_b(n,f)$, we use the same procedure as in Section 4.1. Here, the residual echo $\mathbf{z}(n,f)$ is the only latent signal.

At initialization, we set the residual echo at the value of the echo:

$$\mathbf{z}(n,f) \leftarrow \mathbf{y}(n,f). \tag{196}$$

We initialized the PSDs as

$$v_s(n,f) \leftarrow \frac{1}{M}\|\mathbf{s}(n,f)\|^2 \tag{197}$$

$$v_z(n,f) \leftarrow \frac{1}{M}\|\mathbf{z}(n,f)\|^2 \tag{198}$$

$$v_b(n,f) \leftarrow \frac{1}{M}\|\mathbf{b}(n,f)\|^2 \tag{199}$$

and the SCMs as the identity matrix $\mathbf{I}_M$.

At each iteration, we derive the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in steps 2 and 3 of Fig. 9 respectively. In order to update the residual echo $\mathbf{z}(n,f)$, we apply the echo cancellation filter $\underline{\mathbf{H}}(f)$ to the echo $\mathbf{y}(n,f)$ as in (155).

To obtain the ground truth PSDs $v_s(n,f)$, $v_z(n,f)$ and $v_b(n,f)$, we use a similar procedure as in Section 4.1. In the E-step, the PSDs are computed as

$$v_s(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_s(f)^{-1}\mathbf{s}(n,f)\mathbf{s}(n,f)^H\Big), \tag{200}$$

$$v_z(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_z(f)^{-1}\mathbf{z}(n,f)\mathbf{z}(n,f)^H\Big), \tag{201}$$

$$v_b(n,f) = \frac{1}{M}\mathrm{tr}\Big(\mathbf{R}_b(f)^{-1}\mathbf{b}(n,f)\mathbf{b}(n,f)^H\Big), \tag{202}$$

and the SCMs as

$$\mathbf{R}_s(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_s(n,f)}\mathbf{s}(n,f)\mathbf{s}(n,f)^H, \tag{203}$$

$$\mathbf{R}_z(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_z(n,f)}\mathbf{z}(n,f)\mathbf{z}(n,f)^H, \tag{204}$$

$$\mathbf{R}_b(f) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{v_b(n,f)}\mathbf{b}(n,f)\mathbf{b}(n,f)^H. \tag{205}$$

The SCMs are then normalized so as to encode only the spatial information of the sources as in (193)–(195).

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. 6 proved to provide significant reduction of echo.

The detailed pseudo-code of the ground truth estimation procedure is provided in Alg. 6. After a few iterations, we observed the convergence of the estimated ground truth residual echo $\mathbf{z}(n,f)$. In practice, we found that the signal $\mathbf{z}(n,f)$ derived with this iterative procedure did not change after 3 iterations.

6.4.2 Inputs

We derive the inputs similarly to Section 4.2. The inputs of NN_0 are concatenated as

$$\text{inputs} = \Big[|\widetilde{d}(n,f)|, |x(n,f)|, |\widetilde{y}(n,f)|, |\widetilde{e}(n,f)|, \Big]. \tag{206}$$

For NN_i, we also consider the type-II inputs as additional inputs. The inputs of NN_i are concatenated as

$$\text{inputs} = \left[|\widetilde{d}(n,f)|, |x(n,f)|, |\widetilde{y}(n,f)|, |\widetilde{e}(n,f)|, \left[\sqrt{v_{c'}^{\mathrm{unc}}(n,f))}\right]_{c'\in\mathcal{C}''}\right]. \tag{207}$$

Algorithm 6: Proposed iterative procedure to derive the ground truths PSDs.

Input:
$\mathbf{s}(n,f)$, $\mathbf{y}(n,f)$, $\mathbf{b}(n,f)$
Initialize:
Initialize the latent variables
$\mathbf{z}_r(f) \leftarrow \mathbf{y}(n,f)$
for *each source* $\mathbf{c}$ *of* $[\mathbf{s},\mathbf{z},\mathbf{b}]$ **do**
Initialize the PSDs
$v_c(n,f) \leftarrow$ (197) (198) or (199)
Initialize the SCMs
$\mathbf{R}_c(f) \leftarrow \mathbf{I}_M$
end
for *each iteration* i *of* I **do**
Update the echo cancellation filter
$\underline{\mathbf{h}}(f) \leftarrow$ (177)
Update the latent residual echo
$\mathbf{z}(n,f) \leftarrow$ (155)
for *each source* $\mathbf{c}$ *of* $[\mathbf{s},\mathbf{z},\mathbf{b}]$ **do**
Update the source PSD
$v_c(n,f) \leftarrow$ (200) (201) or (202)
Update the source SCM
$\mathbf{R}_c(f) \leftarrow$ (203)+(193) or (204)+(194) or (205)+(195)
end
end
Output:
$\left[v_s(n,f)\ v_z(n,f)\ v_b(n,f)\right]$

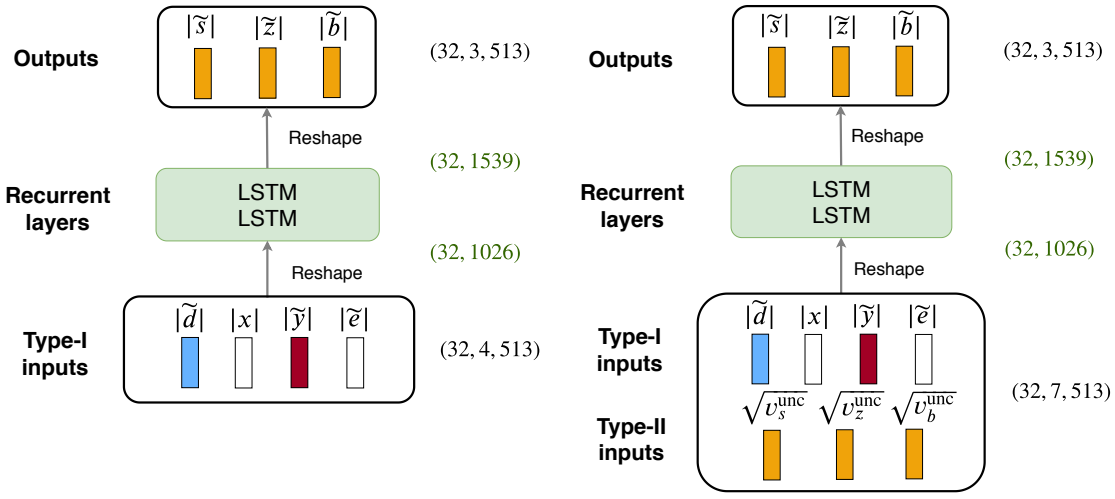
6.4.3 Cost function

Let $|\widetilde{s}(n,f)|$, $|\widetilde{z}(n,f)|$ and $|\widetilde{b}(n,f)|$ denote the NN output for the signals $\mathbf{s}(n,f)$, $\mathbf{z}(n,f)$ and $\mathbf{b}(n,f)$, respectively. As mentioned above, we use NN_0 and NN_i to jointly predict the 3 spectral parameters $\left[|\widetilde{s}(n,f)||\widetilde{z}_{\text{r}}(n,f)||\widetilde{b}(n,f)|\right]$ (see Fig. 10). We use the Kullback-Leibler divergence as the training loss, which has shown to provide the best results for NN training among several other losses [4]:

$$\mathcal{D}_{KL} = \frac{1}{3FN}\sum_{c,n,f}\left(\sqrt{v_c(n,f)}\log\frac{\sqrt{v_c(n,f)}}{|\widetilde{c}(n,f)|} - \sqrt{v_c(n,f)} + |\widetilde{c}(n,f)|\right). \tag{208}$$

6.4.4 Architecture

The neural network follows a similar architecture to the initially-proposed approach for joint reduction of echo, reverberation and noise (see Fig. 10). However, the number of inputs is $4F$ for NN_0 and $7F$ for NN_i. In addition, the number of outputs is $3F$.



(a) NN_0 (b) NN_i, $i \geq 1$

Figure 10: Architecture of the NNs with a sequence length of 32 timesteps and $F = 513$ frequency bins.

7 Experimental protocol

In this section we give details of the recording and simulation parameters for creating the datasets. We also detail the computation of the estimated early near-end components and we recall the baselines.

7.1 Datasets

7.1.1 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.

Dataset		**Training**	**Validation**	**Test**
Signals	$\mathbf{y}$	recorded		recorded
	$\mathbf{s}$	simulated $\mathbf{a}_s$		
	$\mathbf{b}$	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.

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

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. 11) 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.

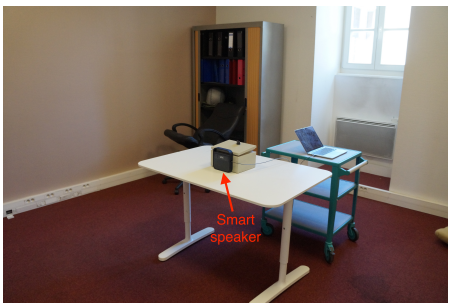


Figure 11: Echo recording setup.

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.

7.1.2 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

RIRs $\mathbf{a}_s(\tau)$ simulated using the Roomsimove toolbox [13]. The simulated rooms matched the same reverberation characteristics (RT_{60} per octave) as the real rooms where the echo recordings were done (see Fig. 12). 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. 13). For each RIR, a random room and a random near-end position were generated to increase the diversity of RIRs.

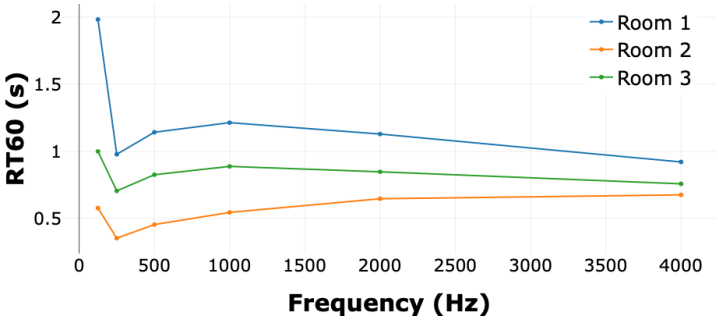


Figure 12: RT_{60} per octave of rooms 1, 2 and 3.

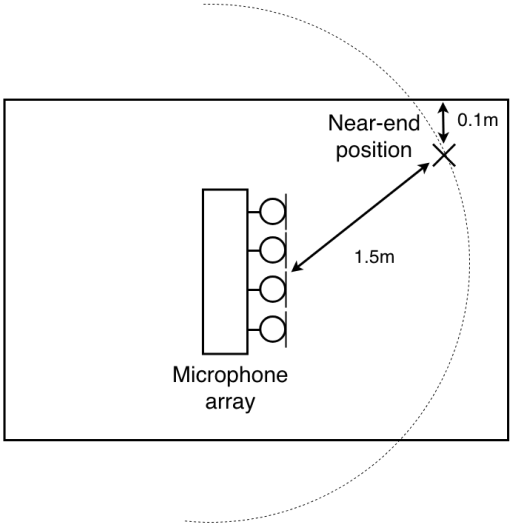


Figure 13: Near-end simulation settings.

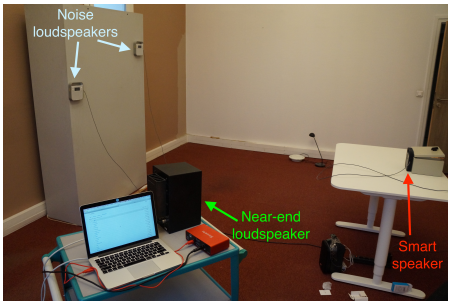


Figure 14: Recording setup for the test set.

7.1.3 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. 12).

7.1.4 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. 14. 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. 14). 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.

7.2 Evaluation metrics

Overall distortion	**SI-SDR**	$10\log_{10}\frac{\|s_{\mathrm{e}}^{\mathrm{post}}\|^2}{\|s_{\mathrm{l}}^{\mathrm{post}}+y^{\mathrm{post}}+b^{\mathrm{post}}+s_{\mathrm{e}}^{\mathrm{art}}\|^2}$
Echo	ERLE	$10\log_{10}\frac{\|y\|^2}{\|y^{\mathrm{post}}\|^2}$
	SER	$10\log_{10}\frac{\|s_{\mathrm{e}}^{\mathrm{post}}\|^2}{\|y^{\mathrm{post}}\|^2}$
Rever-beration	ELR	$10\log_{10}\frac{\|s_{\mathrm{e}}^{\mathrm{post}}\|^2}{\|s_{\mathrm{l}}^{\mathrm{post}}\|^2}$
Noise	SNR	$10\log_{10}\frac{\|s_{\mathrm{e}}^{\mathrm{post}}\|^2}{\|b^{\mathrm{post}}\|^2}$
Artifacts	SI-SAR	$10\log_{10}\frac{\|s_{\mathrm{e}}^{\mathrm{post}}\|^2}{\|s_{\mathrm{e}}^{\mathrm{art}}\|^2}$

Table 3: Evaluation metrics. The formulas are given in the single-channel case ($M = 1$) and the channel index m is omitted for conciseness.

Throughout this subsection, the signals are expressed in the single-channel case ($M = 1$) and the microphone index m is omitted for conciseness. The estimated early near-end signal $\widehat{s}_{\mathrm{e}}(t)$ has 5 components

$$\widehat{s}_{\mathrm{e}}(t) = s_{\mathrm{e}}^{\mathrm{post}}(t) + s_{\mathrm{l}}^{\mathrm{post}}(t) + y^{\mathrm{post}}(t) + b^{\mathrm{post}}(t) + s_{\mathrm{e}}^{\mathrm{art}}(t), \tag{209}$$

where $s_{\mathrm{e}}^{\mathrm{post}}(t)$ is the potentially attenuated early near-end signal, $s_{\mathrm{l}}^{\mathrm{post}}(t)$, $y^{\mathrm{post}}(t)$ and $b^{\mathrm{post}}(t)$ are the post-residual distortion sources that are ideally equal to zero vectors, and $s_{\mathrm{e}}^{\mathrm{art}}(t)$ denotes the artifacts introduced in the early near-end signal $s_{\mathrm{e}}(t)$. We define the four components $s_{\mathrm{e}}^{\mathrm{post}}(t)$, $s_{\mathrm{l}}^{\mathrm{post}}(t)$, $y^{\mathrm{post}}(t)$ and $b^{\mathrm{post}}(t)$ as

$$c^{\mathrm{post}}(t) = \gamma_c \; c(t), \tag{210}$$

where $c(t)$ denotes each of the four original signals $s_{\mathrm{e}}(t)$, $s_{\mathrm{l}}(t)$, y^{post} and b^{post},

$$\gamma_c = \frac{\langle \widehat{s}_{\mathrm{e}}, c \rangle}{\|c\|^2}, \tag{211}$$

$\langle \cdot, \cdot \rangle$ denotes the scalar product of two sampled signals over a fixed time period. Each component

$c^{\text{post}}(t)$ is thus defined as the projection of the estimated early near-end signal $\widehat{s}_{\text{e}}(t)$ on signal $c(t)$. The artifacts are computed as

$$s_{\text{e}}^{\text{art}}(t) = \widehat{s}_{\text{e}}(t) - \left(s_{\text{e}}^{\text{post}}(t) + s_{\text{l}}^{\text{post}}(t) + b^{\text{post}}(t) + y^{\text{post}}(t)\right). \tag{212}$$

Note that the definition of the 5 components of the early near-end signal $\widehat{s}_{\text{e}}(t)$ is an extension of Le Roux et al. in noise reduction to multiple distorsion sources [14].

7.3 Baselines

Hereafter we denote our joint NN-supported approach as *NN-joint*. We compare it with four baselines:

1. *Togami*: our implementation of Togami et al.'s approach [15],

2. *Cascade*: a cascade approach where the echo cancellation filter $\underline{\mathbf{H}}(f)$, the dereverberation filter $\underline{\mathbf{G}}(f)$ and the Wiener postfilter $\mathbf{W}_{s_e}(n, f)$ are estimated and applied one after another. Echo cancellation relies on SpeexDSP[1], which implements Valin's adaptive approach and is particularly suitable for time-varying conditions [11]. Dereverberation relies on our implementation of WPE [2, 6]. The multichannel Wiener postfilter is computed using our implementation of Nugraha et al.'s NN-EM approach [4].

3. *NN-parallel*: the variant of *NN-joint* where the echo cancellation filter $\underline{\mathbf{H}}(f)$ and the dereverberation filter $\underline{\mathbf{G}}(f)$ are applied in parallel as Togami et al.'s approach (see Section 5),

4. *NN-cascade*: the variant of *Cascade* where the echo cancellation filter $\underline{\mathbf{H}}(f)$ is estimated using the NN-supported approach similar to *NN-joint* (see Section 6) instead of Valin's adaptive approach. As WPE dereverberates similarly to its NN-supported counterpart in the multichannel case [16], *NN-cascade* corresponds to a cascade variant of *NN-joint* which estimates each filter separately using NN-supported optimization algorithms.

[1] https://github.com/xiph/speexdsp

8 Results and discussion

In this section we give the results after each filtering step and provides estimated spectrogram examples by all the approaches.

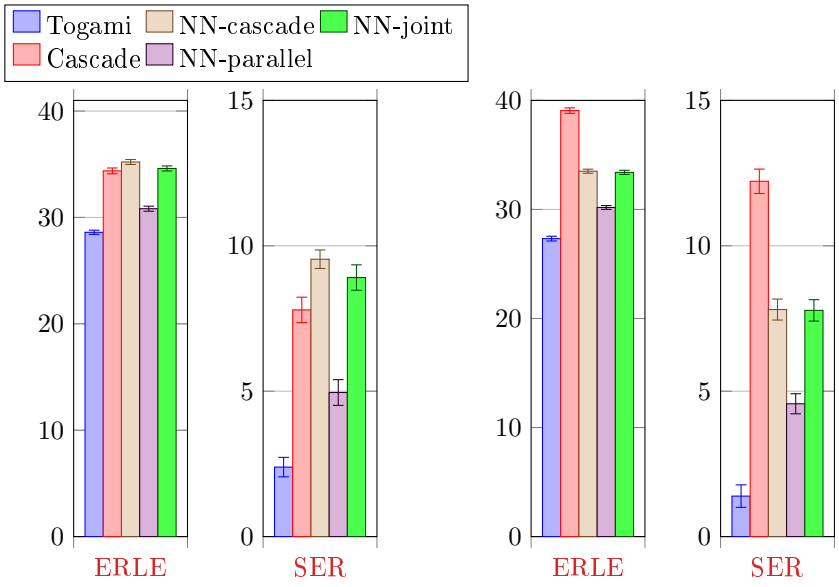
8.1 After linear filtering

Fig. 15a shows the average SER and ERLE after echo cancellation in time-invariant conditions. The other metrics are not provided as echo cancellation only focuses on echo reduction and introduces very few artifacts in the target $\mathbf{s}_\mathrm{e}$ compared to the post-filter $\mathbf{W}_{s_\mathrm{e}}(n, f)$. *NN-joint* outperforms *Togami* in both metrics. *NN-parallel* also outperforms *Togami* in both metrics, but is outperformed by *NN-joint*.

NN-joint performs similarly to *Cascade* in terms of ERLE, and outperforms *Cascade* in terms of SER. *NN-cascade* outperforms *Cascade* in terms of both metrics. However, *NN-joint* and *NN-cascade* perform similarly in terms of SER.

Fig. 15b shows the average SER and ERLE after echo cancellation in time-varying conditions. The trend in performance between *NN-joint*, *NN-parallel* and *Togami* is similar to the average performance in time-invariant conditions. *NN-joint* and *NN-cascade* performs similarly in terms of both metrics. However, *Cascade* significantly outperforms both *NN-joint* and *NN-cascade* in terms of both metrics. This is explained by Valin's adaptive approach for echo cancellation in *Cascade* which is designed for time-varying conditions [11], whereas *NN-joint* and *NN-cascade* are not designed for such conditions.

8.1.1 After echo cancellation



(a) Time-invariant conditions (b) Time-varying conditions

Figure 15: Average results (in dB) after echo cancellation.

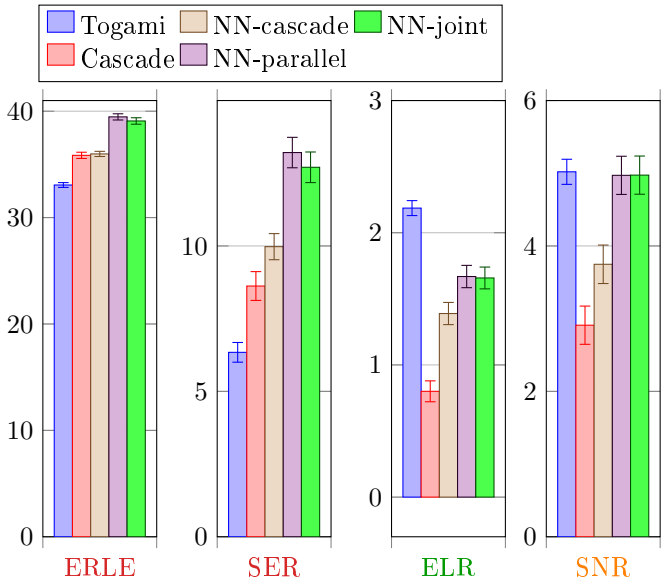
8.1.2 After echo cancellation and linear dereverberation

Fig. 16a shows the average ERLE, SER, ELR and SNR in time-invariant conditions. The other metrics are not provided as linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ introduce very few artifacts in the target $\mathbf{s}_{\mathrm{e}}$ compared to the post-filter $\mathbf{W}_{s_e}(n, f)$. *NN-joint* achieves greater echo reduction than *Togami*, lower dereverberation and similar noise reduction. *NN-parallel* achieves similar performance as *NN-joint*.

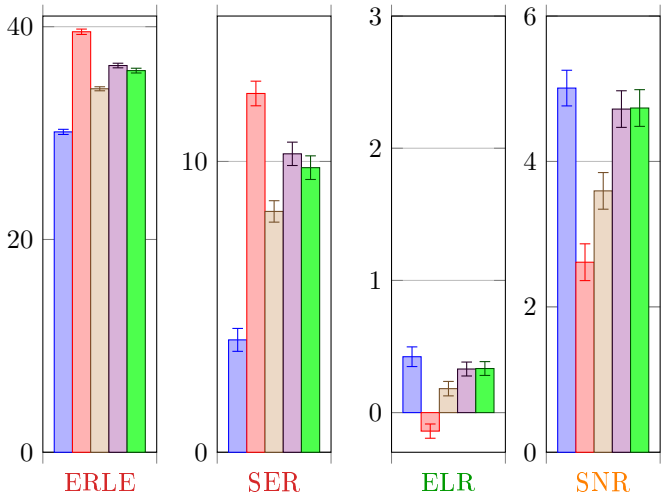
NN-joint outperforms *Cascade* in terms of all metrics. *NN-cascade* also outperforms *Cascade* in terms of all metrics. However, *NN-cascade* is outperformed by *NN-joint* in terms of all metrics. This means that joint estimation of the filters improves performance after applying the two linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$.

Fig. 16b shows the average ERLE, SER, ELR and SNR in time-invariant conditions. The trend in performance is similar to the average performance in time-invariant conditions for all approaches. However, *Cascade* achieves significantly greater echo reduction than all other approches. This is explained by Valin's adaptive approach for echo cancellation in *Cascade* which

is designed for time-varying conditions [11], whereas the other approaches are not designed for such conditions.



(a) Time-invariant conditions



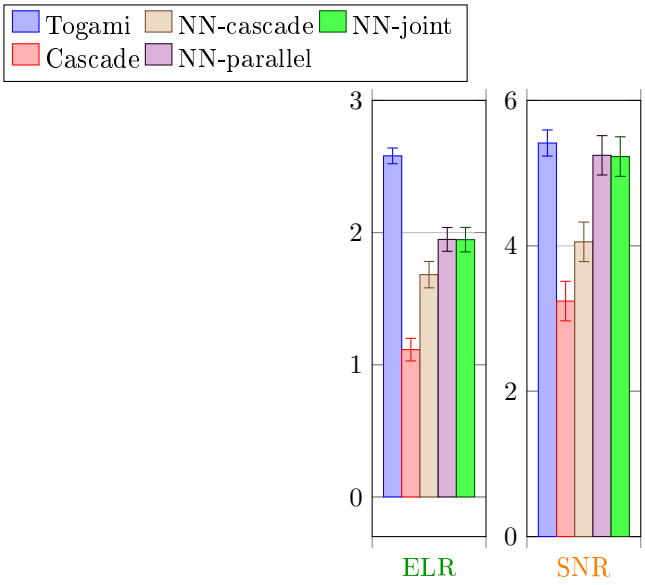
(b) Time-varying conditions

Figure 16: Average results (in dB) after echo cancellation and linear dereverberation.

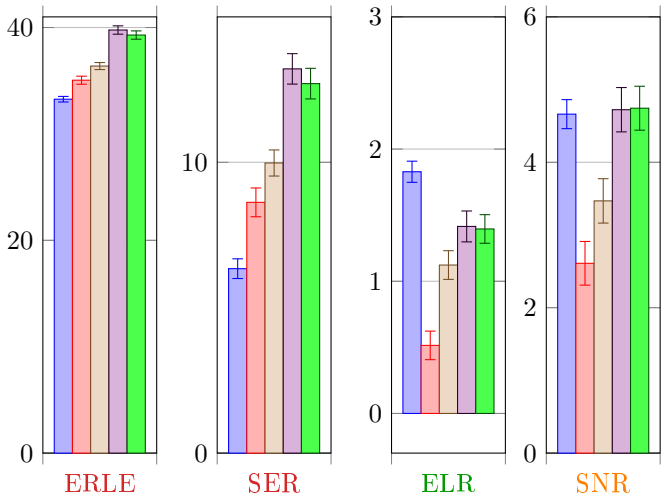
While Fig. 16a shows the performance averaged over all periods (*near-end talk*, *far-end talk* and *double-talk*), we need further performance analysis when only noise and reverberation are present, i.e. during *near-end talk*, and when echo, reverberation and noise are present simulta-

neously, i.e. during *double-talk*, to investigate how the system components interact with each other. We discard the analysis of *far-end talk* as the target $\mathbf{s}_e$ is absent in this scenario.

Fig. 17a shows the average ELR and SNR during *near-end talk*. The trend in performance is similar to the results averaged over all periods. Fig. 17b shows the average ERLE, SER, ELR and SNR during *double-talk*. The trend in performance is similar to the results averaged over all periods.



(a) Time-invariant conditions



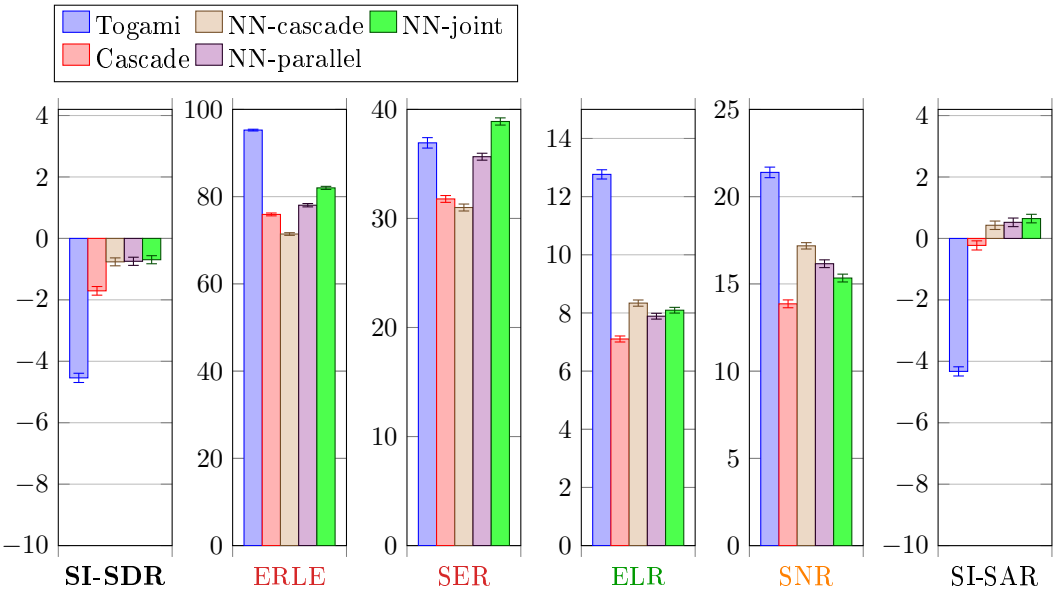
(b) Double-talk

Figure 17: Analysis (in dB) in time-invariant conditions.

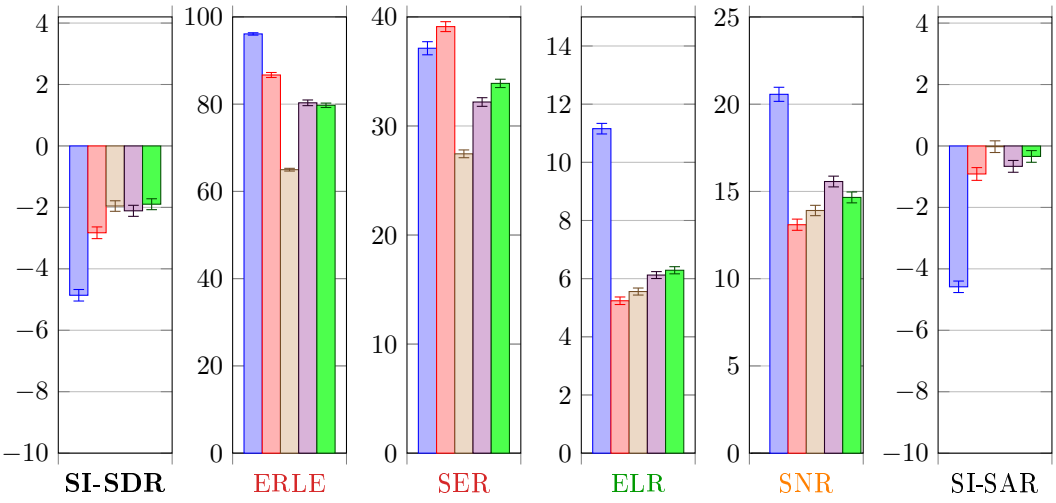
8.2 After post-filtering

Fig. 18a shows the average results in time-invariant conditions. Fig. 18b shows the average results in time-varying conditions. In time-varying conditions, the trend in performance between *NN-joint*, *NN-parallel* and *Togami* is similar to the average performance in time-invariant conditions.

The trend in SI-SDR between *NN-joint*, *NN-cascade* and *Cascade* is similar to the average SI-SDR in time-invariant conditions. However, regarding *NN-joint* and *NN-cascade*, the distribution of the overall distortion is changed: jointly optimizing the filters improves the reduction of echo, reverberation and noise, with similar degradation of the target $\mathbf{s}_e$.

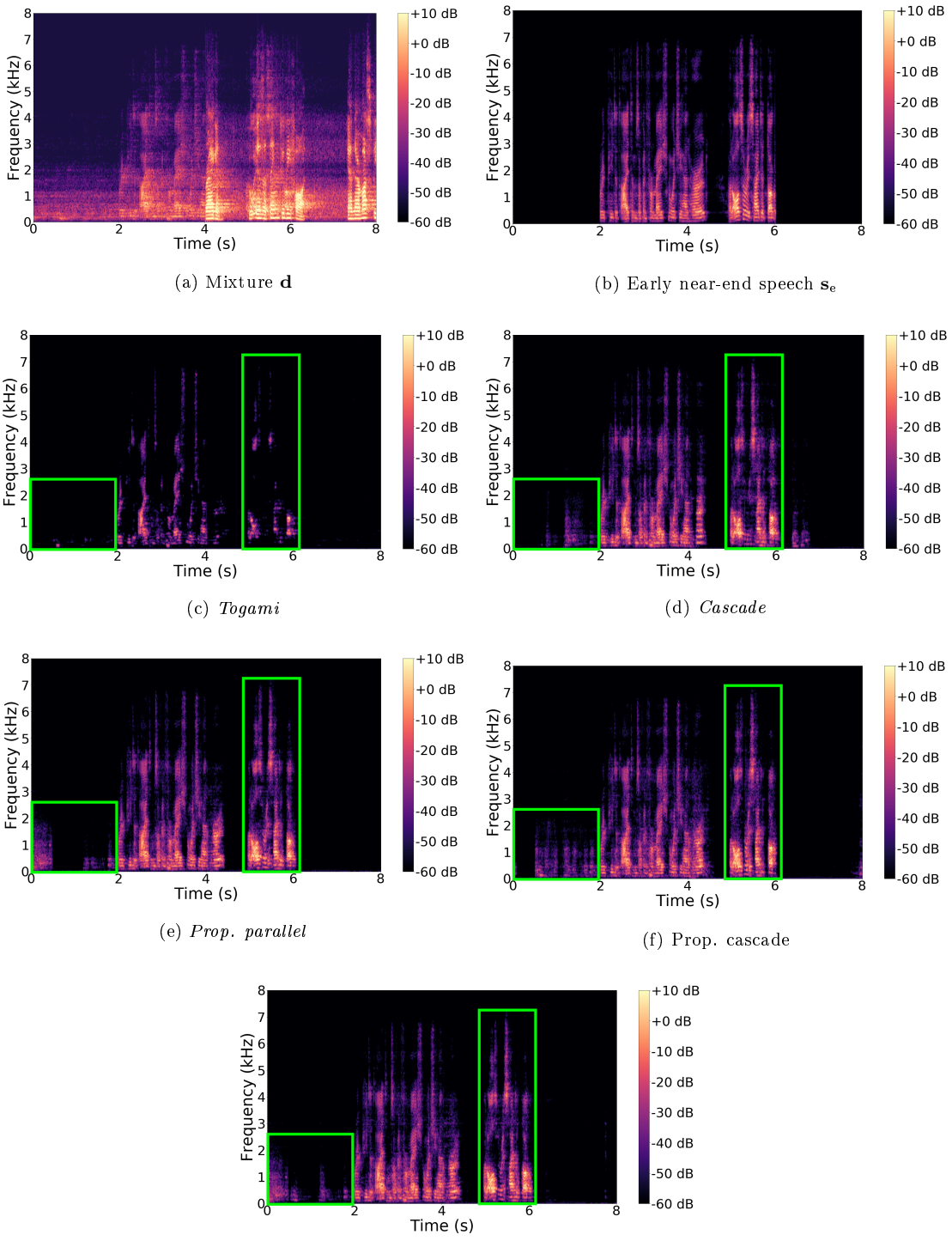


(a) Time-invariant conditions



(b) Time varying conditions

Figure 18: Average results (in dB).



(a) Mixture $\mathbf{d}$

(b) Early near-end speech $\mathbf{s}_e$

(c) *Togami*

(d) *Cascade*

(e) *Prop. parallel*

(f) Prop. cascade

(g) Prop.

Figure 19: Example spectrograms of the estimated early near-end component $\mathbf{s}_e$ with the proposed joint approach and the baselines in time-invariant conditions (only one channel is illustrated). The green rectangles show the areas in the spectrogram where the estimations are different.

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] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Joint DNN-based multichannel reduction of echo, reverberation and noise: Supporting document," Inria, Tech. Rep. RR-9284, 2019.

[11] 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.

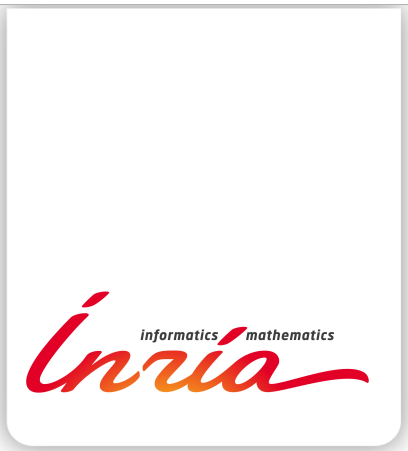
[12] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Multiple-input neural network-based residual echo suppression," in *Proc. ICASSP*, 2018, pp. 231–235.

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

[14] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR — half-baked or well done?" in *Proc. ICASSP*, 2019, pp. 626–630.

[15] M. Togami and Y. Kawaguchi, "Simultaneous optimization of acoustic echo reduction, speech dereverberation, and noise reduction against mutual interference," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 11, pp. 1612–1623, 2014.

[16] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural network-based spectrum estimation for online WPE dereverberation," in *Interspeech*, 2017, pp. 384–388.



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