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

Guillaume Carbajal, Romain Serizel, Emmanuel Vincent, Eric Humbert

▶ **To cite this version:**

Guillaume Carbajal, Romain Serizel, Emmanuel Vincent, Eric Humbert. Joint NN-Supported Multichannel Reduction of Acoustic Echo, Reverberation and Noise. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2020, 10.1109/TASLP.2020.3008974 . hal-02372579v4

HAL Id: hal-02372579

https://inria.hal.science/hal-02372579v4

Submitted on 27 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Guillaume Carbajal, *Student Member, IEEE,* Romain Serizel, *Member, IEEE,* Emmanuel Vincent, *Senior Member, IEEE,* and Eric Humbert, *Member, IEEE*

***Abstract*—We consider the problem of simultaneous reduction of acoustic echo, reverberation and noise. In real scenarios, these distortion sources may occur simultaneously and reducing them implies combining the corresponding distortion-specific filters. As these filters interact with each other, they must be jointly optimized. We propose to model the target and residual signals after linear echo cancellation and dereverberation using a multichannel Gaussian modeling framework and to jointly represent their spectra by means of a neural network. We develop an iterative block-coordinate ascent algorithm to update all the filters. We evaluate our system on real recordings of acoustic echo, reverberation and noise acquired with a smart speaker in various situations. The proposed approach outperforms in terms of overall distortion a cascade of the individual approaches and a joint reduction approach which does not rely on a spectral model of the target and residual signals.**

***Index Terms*—Acoustic echo, reverberation, background noise, joint distortion reduction, expectation-maximization, recurrent neural network.**

I. Introduction

In hands-free telecommunications, a speaker from a near-end point interacts with another speaker at a far-end point. The near-end speaker can be a few meters away from the microphones and the interactions can be subject to several distortion sources such as background noise, acoustic echo and near-end reverberation. Each of these distortion sources degrades speech quality, intelligibility and listening comfort, and must be reduced.

Single- and multichannel filters have been used to reduce each of these distortion sources independently. They can be categorized into short nonlinear filters that vary quickly over time and long linear filters that are time-invariant (or slowly time-varying). Short nonlinear filters are generally used for noise reduction [1]. They are robust to the fluctuations and nonlinearities inherent to real signals. Long linear filters can be required for dereverberation [2] and echo reduction [3]. They are able to reduce most of the distortion sources in time-invariant conditions without introducing any artifact, or musical noise, in the near-end signal.

When several distortion sources occur simultaneously, reducing them requires cascading the distortion-specific filters. However, as these filters interact with each other, tuning them independently can be suboptimal and even lead to additional distortions. Several joint approaches that handle two distortion sources have been proposed, namely for joint dereverberation and source separation/noise reduction [4]–[9], for joint echo and noise reduction [10]–[15], and for joint echo reduction and dereverberation [16], [17].

A joint approach for single-channel echo reduction, dereverberation and noise reduction was proposed by Habets et al. [18]. However, the linear echo cancellation filter was ignored in the optimization process. To the best of our knowledge, only Togami et al. proposed a solution optimizing two linear filters and a nonlinear postfilter for reducing echo, reverberation and noise [19]. They represented the filter interactions by modeling the target and residual signals after echo cancellation and dereverberation within a multichannel Gaussian framework. However, no model was proposed for the short term spectra of these signals. This results in misestimation of the linear filters and the nonlinear postfilter.

Recently, neural networks (NN) have shown promising results to estimate the short term spectra of speech and distortion sources for joint dereverberation and source separation/noise reduction [20], [21], and for joint echo and noise reduction [22], [23]. However, these approaches have only focused on reducing two distortion sources.

In this article, we propose an NN-supported approach for joint multichannel reduction of echo, reverberation and noise. We simultaneously model the spatial and spectral parameters of the target and residual signals within a multichannel Gaussian framework and we derive an iterative a block-coordinate ascent (BCA) algorithm to update the echo cancellation, dereverberation and noise/residual reduction filters. We evaluate our system on real recordings of acoustic echo, near-end reverberation and background noise acquired with a smart speaker in various situations. We experimentally show the effectiveness of our proposed approach compared with a cascade of individual approaches and Togami et al.'s joint reduction approach [19].

The rest of this article is organized as follows. In Section II, we describe existing enhancement methods designed for the separate reduction of echo, reverberation or noise, and Togami et al.'s approach. We explain our joint approach using an NN spectral model within a BCA algorithm in Section III. In Section IV we detail our NN-based joint spectral model. Section V describes the experimental settings for the training and evaluation of our approach. Section VI shows the results of our approach compared to the cascade of individual approaches and Togami et al.'s approach. Finally Section VII concludes the article and provides future directions.

M. Carbajal and E. Humbert are with Invoxia SAS, 8 Esplanade de la Manufacture, 92130 Issy-les-Moulineaux, France (email: guillaume.carbajal@invoxia.com, eric.humbert@invoxia.com). R. Serizel and E. Vincent are with Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France (email: romain.serizel@loria.fr, emmanuel.vincent@inria.fr).

II. Background

In this section, we first describe multichannel approaches for the separate reduction of echo, reverberation or noise. These approaches will be used as building blocks for our solution and a basis for comparison in our experiments. We then describe Togami et al.'s joint approach. We adopt the following notations through the article: 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 as $\mathbf{I}$. Its dimension is either implied by the context or explicitly specified by a subscript.

A. Echo reduction

The echo reduction problem is defined as follows. Denoting by M the number of channels (microphones), the mixture $\mathbf{d}^{\text{echo}}(t) \in \mathbb{R}^{M\times 1}$ observed at the microphones at time t is the sum of the near-end signal $\mathbf{s}(t) \in \mathbb{R}^{M\times 1}$ and the acoustic echo $\mathbf{y}(t) \in \mathbb{R}^{M\times 1}$:

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

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

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

The linear part corresponds to the linear convolution of $x(t)$ and the M-dimensional room impulse response (RIR) $\mathbf{a}_{\text{y}}(\tau) \in \mathbb{R}^{M\times 1}$, or echo path, modeling the acoustic path from the loudspeaker (including the loudspeaker response) to the microphones. The nonlinear part is denoted by $\mathbf{y}_{\text{nl}}(t) \in \mathbb{R}^{M\times 1}$. The signals are transformed into the time-frequency domain by the short-time Fourier transform (STFT)

$$\mathbf{d}^{\text{echo}}(n,f) = \mathbf{s}(n,f) + \mathbf{y}(n,f), \tag{3}$$

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 time frames of the utterance. As the far-end signal $x(n,f) \in \mathbb{C}$ is known, the goal is to recover the M-dimensional near-end speech $\mathbf{s}(n,f) \in \mathbb{C}^{M\times 1}$ from the mixture $\mathbf{d}^{\text{echo}}(n,f) \in \mathbb{C}^{M\times 1}$ by identifying the echo path $\{\mathbf{a}_{\text{y}}(n,f)\}_{n,f}$. The underlying idea is to estimate $\mathbf{y}(n,f) \in \mathbb{C}^{M\times 1}$ with a long, multiframe linear echo cancellation filter $\underline{\mathbf{H}}(f) = [\mathbf{h}(0,f)\dots\mathbf{h}(K-1,f)] \in \mathbb{C}^{M\times K}$ applied on the K previous frames of the far-end signal $x(n,f)$, and to subtract the resulting signal $\widehat{\mathbf{y}}(n,f)$ from $\mathbf{d}^{\text{echo}}(n,f)$:

$$\mathbf{e}^{\text{echo}}(n,f) = \mathbf{d}^{\text{echo}}(n,f) - \underbrace{\sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f)}_{=\widehat{\mathbf{y}}(n,f)}. \tag{4}$$

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 the tap k is measured in frames and the underscore notation in $\underline{\mathbf{H}}(f)$ denotes the concatenation of the K taps of $\mathbf{h}(k,f)$. Since the far-end signal $x(n,f)$ is known, the filter $\underline{\mathbf{H}}(f)$ is usually estimated adaptively in the minimum mean square error (MMSE) sense [3]. Adaptive MMSE optimization typically relies on adaptive algorithms such as least mean squares (LMS) which adjust the filter $\underline{\mathbf{H}}(f)$ in an online manner by stochastic gradient descent with a time-varying step size [3]. These algorithms have low complexity and fast convergence, which makes them particularly suitable for time-varying conditions. Yang et al. provide a comprehensive review of optimal step size selection for echo reduction [24].

In practice, the output signal $\mathbf{e}^{\text{echo}}(n,f)$ is not equal to the near-end speech $\mathbf{s}(n,f)$, not only because of the estimation error, but also because of the smaller length of $\underline{\mathbf{H}}(f)$ compared to the true echo path and of nonlinearities $\mathbf{y}_{\text{nl}}(n,f)$ that cannot be modeled by $\underline{\mathbf{H}}(f)$ [18]. As a result, a residual echo $\mathbf{z}(n,f)$ remains that can be expressed as [3]

$$\mathbf{e}^{\text{echo}}(n,f) - \mathbf{s}(n,f) = \underbrace{\mathbf{y}(n,f) - \widehat{\mathbf{y}}(n,f)}_{=\mathbf{z}(n,f)}. \tag{5}$$

To overcome this limitation, a (nonlinear) residual echo suppression postfilter $\mathbf{W}^{\text{echo}}(n,f) \in \mathbb{C}^{M\times M}$ is typically applied:

$$\widehat{\mathbf{s}}(n,f) = \mathbf{W}^{\text{echo}}(n,f)\mathbf{e}^{\text{echo}}(n,f). \tag{6}$$

There exist multiple approaches to derive $\mathbf{W}^{\text{echo}}(n,f)$ [3]. Recently, direct estimation of $\mathbf{W}^{\text{echo}}(n,f)$ using an NN has shown good performance in the single-channel case [25], [26]. However, when $\underline{\mathbf{H}}(f)$ changes, $\mathbf{z}(n,f)$ also changes and the postfilter $\mathbf{W}^{\text{echo}}(n,f)$ must be adapted consequently. Estimating $\underline{\mathbf{H}}(f)$ and $\mathbf{W}^{\text{echo}}(n,f)$ separately is thus suboptimal. Joint optimization of $\underline{\mathbf{H}}(f)$ and $\mathbf{W}^{\text{echo}}(n,f)$ was investigated in the MMSE and the maximum likelihood (ML) sense [27], [28].

In Section V, we will use adaptive MMSE optimization for estimating the echo cancellation filter $\underline{\mathbf{H}}(f)$ as a part of the cascade of the individual approaches to which we compare our joint approach.

B. Near-end dereverberation

The near-end dereverberation problem is defined as follows. The signal $\mathbf{d}^{\text{rev}}(t)$ observed at the microphones at time t is just the reverberant near-end signal $\mathbf{s}(t)$, which is obtained by linear convolution of the anechoic near-end signal $u(t) \in \mathbb{R}$ and the M-dimensional RIR $\mathbf{a}_{\text{s}}(\tau) \in \mathbb{R}^{M\times 1}$:

$$\mathbf{d}^{\text{rev}}(t) = \mathbf{s}(t) = \sum_{\tau=0}^{\infty} \mathbf{a}_{\text{s}}(\tau)u(t-\tau). \tag{7}$$

This signal can be decomposed as

$$\mathbf{s}(t) = \underbrace{\sum_{0\le\tau\le t_{\text{e}}} \mathbf{a}_{\text{s}}(\tau)u(t-\tau)}_{=\mathbf{s}_{\text{e}}(t)} + \underbrace{\sum_{\tau>t_{\text{e}}} \mathbf{a}_{s}(\tau)u(t-\tau)}_{=\mathbf{s}_{\text{l}}(t)}, \tag{8}$$

where $\mathbf{s}_{\text{e}}(t)$ denotes the early near-end signal component, $\mathbf{s}_{\text{l}}(t)$ the late reverberation component, and t_{e} is the mixing time. The component $\mathbf{s}_{\text{e}}(t)$ comprises the main peak of the RIR (the direct path) 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. In the time-frequency domain, the reverberant near-end speech can thus be expressed as

$$\mathbf{s}(n,f) = \mathbf{s}_{\mathrm{e}}(n,f) + \mathbf{s}_{\mathrm{l}}(n,f). \tag{9}$$

The goal is to recover the early near-end component $\mathbf{s}_{\mathrm{e}}(n,f)$ from the reverberant near-end signal $\mathbf{s}(n,f)$. Naylor et al. provided a comprehensive review of dereverberation approaches [2]. Among them, the weighted prediction error (WPE) method [29] estimates $\mathbf{s}_{\mathrm{l}}(n,f)$ by inverse filtering with a long, multi-frame linear filter $\underline{\mathbf{G}}(f) = [\mathbf{G}(\Delta,f)\dots\mathbf{G}(\Delta+L-1,f)] \in \mathbb{C}^{M\times ML}$ applied on the L previous frames of the mixture signal $\mathbf{s}(n-\Delta,f)$ defined in (7). The delay Δ is introduced to avoid over-whitening of the near-end speech. The resulting signal $\widehat{\mathbf{s}}_{\mathrm{l}}(n,f)$ is then subtracted from the mixture signal $\mathbf{s}(n,f)$ defined in (7):

$$\mathbf{r}^{\mathrm{rev}}(n,f) = \mathbf{s}(n,f) - \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{s}(n-l,f)}_{\widehat{\mathbf{s}}_{\mathrm{l}}(n,f)}. \tag{10}$$

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 $\underline{\mathbf{G}}(f)$, and $\mathbf{g}_m(l,f) \in \mathbb{C}^{M\times 1}$ is the m-th channel vector of $\mathbf{G}(l,f)$. As the component $\mathbf{s}_{\mathrm{e}}(n,f)$ is not an observed signal, Nakatani et al. estimated the filter $\underline{\mathbf{G}}(f)$ in the ML sense by modeling $\mathbf{s}_{\mathrm{e}}(n,f)$ as a directional source [29]. However, they did not impose any constraint on its short-term spectrum which results in limited dereverberation [29], [30]. Other authors have assumed a model of the short-term spectrum. Yoshioka et al. used an all-pole model [8], Kagami et al. used nonnegative matrix factorization (NMF) [9], Jukić et al. used sparse priors [31], and Kinoshita et al. used an NN [32].

For several reasons, including the smaller length of the filter compared to true near-end RIR and potentially time-varying conditions, a residual late reverberation component $\mathbf{s}_{\mathrm{r}}(n,f)$ remains [33]–[35] that can be expressed as

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

To overcome this limitation, a (nonlinear) residual reverberation suppression postfilter $\mathbf{W}^{\mathrm{rev}}(n,f) \in \mathbb{C}^{M\times M}$ is applied on the signal $\mathbf{r}^{\mathrm{rev}}(n,f)$:

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f) = \mathbf{W}^{\mathrm{rev}}(n,f)\mathbf{r}^{\mathrm{rev}}(n,f). \tag{12}$$

There are multiple approaches to derive $\mathbf{W}^{\mathrm{rev}}(n,f)$ [33], [35]. However, when $\underline{\mathbf{G}}(f)$ changes, $\mathbf{s}_{\mathrm{r}}(n,f)$ also changes and the postfilter $\mathbf{W}^{\mathrm{rev}}(n,f)$ must be adapted consequently. Estimating $\underline{\mathbf{G}}(f)$ and $\mathbf{W}^{\mathrm{rev}}(n,f)$ separately is thus suboptimal. Joint optimization of $\underline{\mathbf{G}}(f)$ and $\mathbf{W}^{\mathrm{rev}}(n,f)$ was investigated in the ML sense [34].

In section V, we will use WPE for estimating the dereverberation filter $\underline{\mathbf{G}}(f)$ as a part of the cascade of the individual approaches to which we compare our joint approach.

C. Noise reduction

The noise reduction problem is defined as follows. In the time-frequency domain, the M-channel mixture $\mathbf{d}^{\mathrm{noise}}(n,f)$ observed at the microphones is the sum of the near-end signal $\mathbf{s}(n,f)$ and a noise signal $\mathbf{b}(n,f) \in \mathbb{C}^{M\times 1}$:

$$\mathbf{d}^{\mathrm{noise}}(n,f) = \mathbf{s}(n,f) + \mathbf{b}(n,f). \tag{13}$$

Note that the noise signal $\mathbf{b}(n,f)$ can be either spatially diffuse or localized. The goal is to recover the near-end speech $\mathbf{s}(n,f)$ from the mixture $\mathbf{d}^{\mathrm{noise}}(n,f)$. This is typically achieved by applying a short nonlinear filter $\mathbf{W}_s^{\mathrm{noise}}(n,f) \in \mathbb{C}^{M\times M}$ on $\mathbf{d}^{\mathrm{noise}}(n,f)$:

$$\widehat{\mathbf{s}}(n,f) = \mathbf{W}^{\mathrm{noise}}(n,f)\mathbf{d}^{\mathrm{noise}}(n,f). \tag{14}$$

The filter can be estimated in the MMSE or ML sense. Gannot el al. provide a comprehensive review of spatial filtering solutions [36]. One family of solutions relies on multichannel time-varying Wiener filtering, where the filter is derived from a local Gaussian model of the target and noise sources [37]. The spectral parameters (short term power spectra) and the spatial parameters (spatial covariance matrices) of this model are estimated in the ML sense. Since there is no closed form solution, the ML parameters are estimated using an EM algorithm. When no constraint is imposed on the spectral or spatial parameters, the EM algorithm operates in each frequency bin f independently which results in a permutation ambiguity in the separated components at each frequency bin f and requires additional permutation alignment. Alternatively, the spectral parameters can be estimated with a model. Ozerov et al. used NMF [38], Nugraha et al. used an NN [39], and recently variational autoencoders were used [40].

In Section V, we will use multichannel time-varying Wiener filtering as a part of the cascade of the individual approaches to which we compare our joint approach.

D. Joint reduction of echo, reverberation and noise

In real scenarios, all the distortions introduced above can be simultaneously present as illustrated in Fig. 1. The mixture $\mathbf{d}(n,f)$ observed at the microphones is thus the sum of the acoustic echo $\mathbf{y}(n,f)$, the reverberant near-end signal $\mathbf{s}(n,f)$ and the noise $\mathbf{b}(n,f)$

$$\mathbf{d}(n,f) = \mathbf{s}(n,f) + \mathbf{y}(n,f) + \mathbf{b}(n,f) \tag{15}$$

$$= \mathbf{s}_{\mathrm{e}}(n,f) + \mathbf{s}_{\mathrm{l}}(n,f) + \mathbf{y}(n,f) + \mathbf{b}(n,f). \tag{16}$$

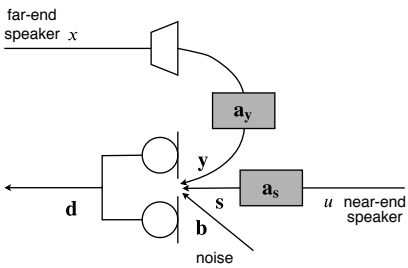


Fig. 1: Acoustic echo, reverberation and noise problem.

The goal is to recover the early near-end component $\mathbf{s}_{\mathrm{e}}(n,f)$ from the mixture $\mathbf{d}(n,f)$.

Togami et al. proposed a joint approach combining an echo cancellation filter $\underline{\mathbf{H}}(f)$ (see Section II-A), a dereverberation filter $\underline{\mathbf{G}}(f)$ (see Section II-B), and a nonlinear multichannel Wiener postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ (see Section II-C) [19]. The approach is illustrated in Fig. 2. In the first step, they apply the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in (4) and subtract the resulting echo estimate $\widehat{\mathbf{y}}(n,f)$ from the mixture signal $\mathbf{d}(n,f)$. In parallel, the authors apply the dereverberation filter $\underline{\mathbf{G}}(f)$ on the mixture signal $\mathbf{d}(n,f)$ as in (10) 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

$$\mathbf{r}(n,f) = \mathbf{d}(n,f) - \widehat{\mathbf{y}}(n,f) - \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{d}(n-l,f)}_{=\widehat{\mathbf{d}}_{\mathrm{l}}(n,f)}. \quad (17)$$

Due to the reasons mentioned in Sections II-A and II-B, and to the presence of the noise signal $\mathbf{b}(n,f)$, undesired residual signals remain and can be expressed as

$$\mathbf{r}(n,f) - \mathbf{s}_{\mathrm{e}}(n,f) = \mathbf{z}_{\mathrm{e}}(n,f) + \widetilde{\mathbf{b}}_{\mathrm{r}}(n,f) + \mathbf{b}_{\mathrm{r}}(n,f). \quad (18)$$

The signals $\mathbf{z}_{\mathrm{e}}(n,f)$, $\widetilde{\mathbf{b}}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ are defined as

$$\mathbf{z}_{\mathrm{e}}(n,f) = \mathbf{y}_{\mathrm{e}}(n,f) - \widehat{\mathbf{y}}(n,f), \quad (19)$$

$$\widetilde{\mathbf{b}}_{\mathrm{r}}(n,f) = \mathbf{s}_{\mathrm{l}}(n,f) - \widehat{\mathbf{d}}_{\mathrm{l},s}(n,f) + \mathbf{y}_{\mathrm{l}}(n,f) - \widehat{\mathbf{d}}_{\mathrm{l},y}(n,f), \quad (20)$$

$$\mathbf{b}_{\mathrm{r}}(n,f) = \mathbf{b}(n,f) - \widehat{\mathbf{d}}_{\mathrm{l},b}(n,f), \quad (21)$$

where the signals $\mathbf{y}_{\mathrm{e}}(n,f)$ and $\mathbf{y}_{\mathrm{l}}(n,f)$ denote the early component and the late reverberation of the echo $\mathbf{y}(n,f)$, respectively, $\widehat{\mathbf{d}}_{\mathrm{l},s}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{s}(n-l,f)$, $\widehat{\mathbf{d}}_{\mathrm{l},y}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{y}(n-l,f)$ and $\widehat{\mathbf{d}}_{\mathrm{l},b}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{b}(n-l,f)$ are the latent components of $\widehat{\mathbf{d}}_{\mathrm{l}}(n,f)$ resulting from (17), and $\mathbf{b}_{\mathrm{r}}(n,f)$ is the *dereverberated* noise signal. The term *dereverberated* means "after applying the dereverberation filter".

To recover the early near-end signal component $\mathbf{s}_{\mathrm{e}}(n,f)$ from the signal $\mathbf{r}(n,f)$, the authors applied a multichannel Wiener postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f) \in \mathbb{C}^{M\times M}$ on the signal $\mathbf{r}(n,f)$:

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f) = \mathbf{W}_{s_{\mathrm{e}}}(n,f)\mathbf{r}(n,f). \quad (22)$$

The authors estimated $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ by modeling $\mathbf{s}_{\mathrm{e}}(n,f)$ and $\mathbf{b}_r(n,f)$ as zero-mean multichannel Gaussian variables, and $\mathbf{z}_{\mathrm{e}}(n,f)$ and $\widetilde{\mathbf{b}}_r(n,f)$ as nonzero-mean multichannel Gaussian variables [19]. They used an EM algorithm to jointly optimize the spectral and spatial parameters of this model in the ML sense.

However, their approach suffers from several limitations. First, they did not impose any constraint on the spectral parameters of the target $\mathbf{s}_{\mathrm{e}}(n,f)$ and the *dereverberated* noise signal $\mathbf{b}_{\mathrm{r}}(n,f)$. Secondly, the signal components $\mathbf{s}_{\mathrm{l}}(n,f)$ and $\mathbf{y}_{\mathrm{l}}(n,f)$ in $\widetilde{\mathbf{b}}_{\mathrm{r}}(n,f)$ are not separately modeled, i.e. these components share the same spatial parameters, which is not the case in practice. These two limitations result in misestimation

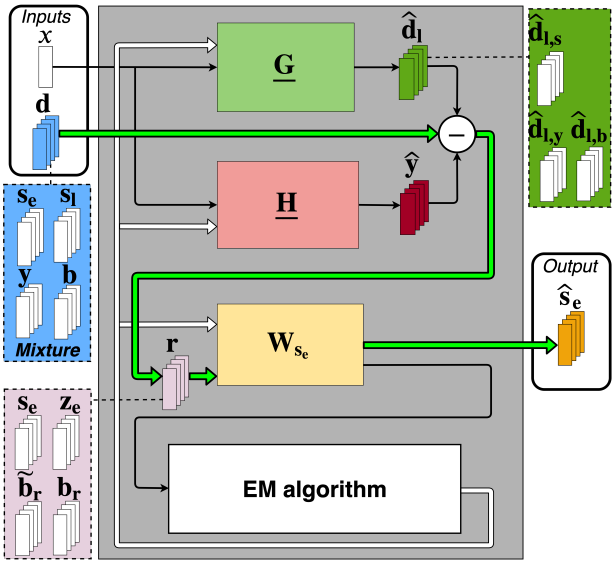


Fig. 2: Togami et al.'s approach for joint reduction of echo, reverberation and noise [19]. The bold green arrows denote the filtering steps. The dashed lines denote the latent signal components. The thin black arrows denote the signals used for the filtering steps and for the joint update. The bold white arrows denote the filter updates.

of the filters $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and the postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$. Thirdly, because the filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ operate independently on the mixture signal $\mathbf{d}(n,f)$, their respective components $\widehat{\mathbf{y}}(n,f)$ and $\widehat{\mathbf{d}}_{\mathrm{l},y}(n,f)$ subtracted from the echo $\mathbf{y}(n,f)$ in (19) and (20) might interfere with each other. Finally, since the echo $\mathbf{y}(n,f)$ is often much louder than the near-end speech $\mathbf{s}(n,f)$ and the noise signal $\mathbf{b}(n,f)$ in $\mathbf{d}(n,f)$, the dereverberation filter $\underline{\mathbf{G}}(f)$ here mainly reduces the late reverberation of the echo $\mathbf{y}_{\mathrm{l}}(n,f)$ instead of the reverberation of the near-end speech $\mathbf{s}_{\mathrm{l}}(n,f)$.

III. NN-SUPPORTED BCA ALGORITHM FOR JOINT REDUCTION OF ECHO, REVERBERATION AND NOISE

In this section, we propose a joint NN-supported model to estimate the spectral parameters of the target and the residual signals. We derive an NN-supported BCA algorithm for joint reduction of echo, reverberation and noise that exploits these estimated spectral parameters for an accurate derivation of the echo cancellation and dereverberation filters and the nonlinear postfilter.

A. Model

The approach is illustrated in Fig. 3. In the first step, we apply the echo cancellation filter $\underline{\mathbf{H}}(f)$ as in (4) and subtract the resulting echo estimate $\widehat{\mathbf{y}}(n,f)$ from $\mathbf{d}(n,f)$:

$$\mathbf{e}(n,f) = \mathbf{d}(n,f) - \underbrace{\sum_{k=0}^{K-1} \mathbf{h}(k,f)x(n-k,f)}_{=\widehat{\mathbf{y}}(n,f)}. \quad (23)$$

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

$\mathbf{b}(n,f)$. Unlike Togami et al. [19], we do not apply the dereverberation filter $\underline{\mathbf{G}}(f)$ on the mixture signal $\mathbf{d}(n,f)$, but on the signal $\mathbf{e}(n,f)$ and subtract the resulting late reverberation estimate $\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)$ from $\mathbf{e}(n,f)$. To the best of our knowledge, this is the first work where the dereverberation filter $\underline{\mathbf{G}}(f)$ is applied after the echo cancellation filter $\underline{\mathbf{H}}(f)$ in the context of joint echo reduction of echo, reverberation and noise. The resulting signal $\mathbf{r}(n,f)$ is thus expressed as

$$\mathbf{r}(n,f) = \mathbf{e}(n,f) - \underbrace{\sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{e}(n-l,f)}_{=\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)}. \tag{24}$$

Since the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are causal, we make the assumption that the observed signals $\mathbf{d}(n,f)$ and $x(n,f)$ are equal to zero for $n<0$. Since the residual echo $\mathbf{z}(n,f)$ in $\mathbf{e}(n,f)$ is a reduced version of the echo $\mathbf{y}(n,f)$ in $\mathbf{d}(n,f)$, the dereverberation filter $\underline{\mathbf{G}}(f)$ should achieve a greater reduction of the near-end late reverberation $\mathbf{s}_{\mathrm{l}}(n,f)$ than in Togami et al.'s approach [19]. Due to the reasons mentioned in Sections II-A and II-B, and to the presence of the noise signal $\mathbf{b}(n,f)$, undesired residual signals remain and can be expressed as

$$\mathbf{r}(n,f) - \mathbf{s}_{\mathrm{e}}(n,f) = \mathbf{s}_{\mathrm{r}}(n,f) + \mathbf{z}_{\mathrm{r}}(n,f) + \mathbf{b}_{\mathrm{r}}(n,f), \tag{25}$$

where $\mathbf{s}_{\mathrm{r}}(n,f)$ is the residual late reverberation near-end component (see Section II-B), $\mathbf{z}_{\mathrm{r}}(n,f)$ is the *dereverberated* residual echo which represents the residual echo remaining after linear dereverberation reduced its linear component (see Section II-A), and $\mathbf{b}_{\mathrm{r}}(n,f)$ is the *dereverberated* noise which represents the residual noise remaining after linear dereverberation reduced its stationary component. The signals $\mathbf{s}_{\mathrm{r}}(n,f)$, $\mathbf{z}_{\mathrm{r}}(n,f)$ and $\mathbf{b}_{\mathrm{r}}(n,f)$ are defined as

$$\mathbf{s}_{\mathrm{r}}(n,f) = \mathbf{s}_{\mathrm{l}}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},s}(n,f), \tag{26}$$
$$\mathbf{z}_{\mathrm{r}}(n,f) = \mathbf{z}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},z}(n,f), \tag{27}$$
$$\mathbf{b}_{\mathrm{r}}(n,f) = \mathbf{b}(n,f) - \widehat{\mathbf{e}}_{\mathrm{l},b}(n,f), \tag{28}$$

where the signals $\widehat{\mathbf{e}}_{\mathrm{l},s}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{s}(n-l,f)$, $\widehat{\mathbf{e}}_{\mathrm{l},z}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{z}(n-l,f)$ and $\widehat{\mathbf{e}}_{\mathrm{l},b}(n,f) = \sum_{l=\Delta}^{\Delta+L-1} \mathbf{G}(l,f)\mathbf{b}(n-l,f)$ are the latent components of $\widehat{\mathbf{e}}_{\mathrm{l}}(n,f)$ resulting from (24). To recover the signal $\mathbf{s}_{\mathrm{e}}(n,f)$ from the signal $\mathbf{r}(n,f)$, we apply a multichannel Wiener postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f) \in \mathbb{C}^{M\times M}$ on the signal $\mathbf{r}(n,f)$ as

$$\widehat{\mathbf{s}}_{\mathrm{e}}(n,f) = \mathbf{W}_{s_{\mathrm{e}}}(n,f)\mathbf{r}(n,f). \tag{29}$$

Inspired by WPE for dereverberation [29], we estimate $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{G}}(f)$ and $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ 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. In the following we use the general notation $\mathbf{c}(n,f)$ to denote each one of these four signals, and consider them as *sources* to be separated. Each of these four sources is modeled as

$$\mathbf{c}(n,f) \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, v_c(n,f)\mathbf{R}_c(f)\right), \tag{30}$$

where $v_c(n,f) \in \mathbb{R}_+$ and $\mathbf{R}_c(f) \in \mathbb{C}^{M\times M}$ denote the power spectral density (PSD) and the spatial covariance matrix (SCM) of the source, respectively [37]. The multichannel

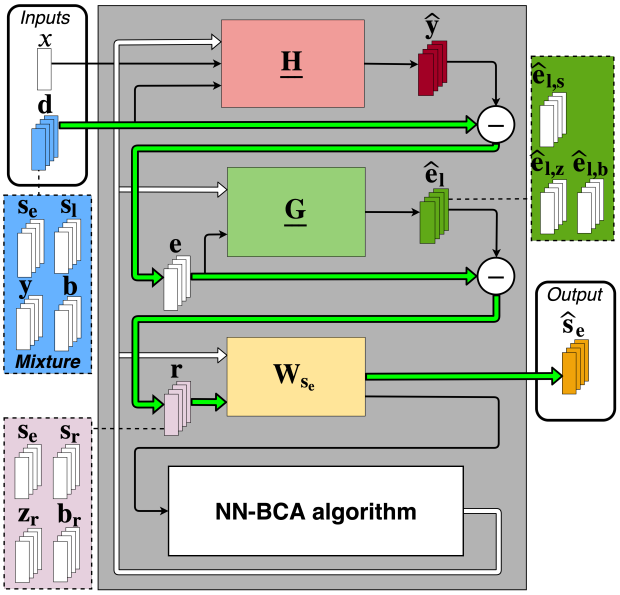


Fig. 3: Proposed approach. Arrows and lines have the same meaning as in Fig. 2.

Wiener filter for the source $\mathbf{c}(n,f)$ is formulated as

$$\mathbf{W}_c(n,f) = v_c(n,f)\mathbf{R}_c(f)\left(\sum_{c'\in\mathcal{C}} v_{c'}(n,f)\mathbf{R}_{c'}(f)\right)^{-1}, \tag{31}$$

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 $\mathbf{r}(n,f)$. The postfilter $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ is a specific case of (31) where $\mathbf{c}(n,f) = \mathbf{s}_{\mathrm{e}}(n,f)$.

B. Likelihood

In order to estimate the parameters of this model, we must first express its likelihood. Following (23), (24), (25) and (30), the log-likelihood of the observed sequence $\mathcal{O} = \{\mathbf{d}(n,f), x(n,f)\}_{n,f}$ is given by

$$\begin{aligned}\mathcal{L}\Big(\mathcal{O};\Theta_H,\Theta_G,\Theta_c\Big) &= \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log p\Big(\mathbf{d}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f), \\ &\qquad x(n,f),\ldots,x(0,f)\Big), \qquad (32)\\ &= \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} \log \mathcal{N}_{\mathbb{C}}\Big(\mathbf{d}(n,f);\boldsymbol{\mu}_{\mathbf{d}}(n,f),\mathbf{R}_{\mathbf{dd}}(n,f)\Big), \qquad (33)\end{aligned}$$

where

$$\boldsymbol{\mu}_{\mathbf{d}}(n,f) = \sum_{k=0}^{K-1}\mathbf{h}(k,f)x(n-k,f) + \sum_{l=\Delta}^{\Delta+L-1}\mathbf{G}(l,f)\mathbf{e}(n-l,f), \tag{34}$$

$$\mathbf{R}_{\mathbf{dd}}(n,f) = \sum_{c'\in\mathcal{C}} v_{c'}(n,f)\mathbf{R}_{c'}(f), \tag{35}$$

and $\Theta_H = \{\underline{\mathbf{H}}(f)\}_f$, $\Theta_G = \{\underline{\mathbf{G}}(f)\}_f$ and $\Theta_c = \{v_c(n,f), \mathbf{R}_c(f)\}_{c,n,f}$ are the parameters to be estimated. The resulting ML optimization problem has no closed form

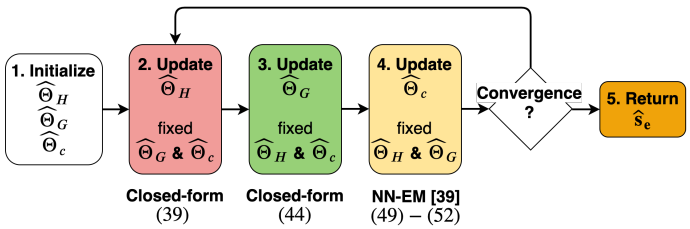


Fig. 4: Flowchart of the proposed BCA algorithm.

solution, hence we need to estimate the parameters via an iterative procedure.

C. Iterative optimization algorithm

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

$$\widehat{\Theta}_H \leftarrow \underset{\Theta_H}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \Theta_H, \widehat{\Theta}_G, \widehat{\Theta}_c\right), \tag{36}$$

$$\widehat{\Theta}_G \leftarrow \underset{\Theta_G}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \Theta_G, \widehat{\Theta}_c\right), \tag{37}$$

$$\widehat{\Theta}_c \leftarrow \underset{\Theta_c}{\operatorname{argmax}}\, \mathcal{L}\left(\mathcal{O}; \widehat{\Theta}_H, \widehat{\Theta}_G, \Theta_c\right). \tag{38}$$

The solutions of (36) and (37) are closed-form. As there is no closed-form solution for (38), we propose to use a modified version of Nugraha et al.'s NN-EM algorithm [39]. The overall flowchart of the proposed algorithm is shown in Fig. 4. Note that it is also possible to optimize the parameters Θ_H, Θ_G and Θ_c with the EM algorithm by adding a nuisance term to (25) [38]. However, this approach would be less efficient to derive the filter parameters Θ_H and Θ_G. In the next subsections, we provide the initialization and the update rules for steps (36)–(38) of our proposed algorithm at iteration i. The derivation of these update rules is detailed in our companion technical report [41, Sec. 3]. At each iteration i, we use the dereverberation filter parameters Θ_G and the source parameters Θ_c of the previous iteration $i-1$.

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 $v_c(n,f)$ of the four sources are jointly initialized using a pretrained NN denoted as NN_0 and the SCMs $\mathbf{R}_c(f)$ as the identity matrix $\mathbf{I}_M$. The inputs, the targets and the architecture of NN_0 are described in Section IV below.

2) Echo cancellation filter parameters Θ_H: The echo cancellation filter $\underline{\mathbf{H}}(f)$ is updated as

$$\underline{\mathbf{h}}(f) = \mathbf{P}(f)^{-1}\mathbf{p}(f), \tag{39}$$

where

$$\mathbf{P}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}_{\text{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{X}}_{\text{r}}(n,f), \tag{40}$$

$$\mathbf{p}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{X}}_{\text{r}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \mathbf{r}_d(n,f). \tag{41}$$

$\underline{\mathbf{h}}(f) = [\mathbf{h}(0,f)^T \dots \mathbf{h}(K-1,f)^T]^T \in \mathbb{C}^{MK\times 1}$ is a vectorized version of $\underline{\mathbf{H}}(f)$, $\underline{\mathbf{X}}_{\text{r}}(n,f) = [\mathbf{X}_{\text{r}}(n,f) \dots \mathbf{X}_{\text{r}}(n-K+1,f)] \in \mathbb{C}^{M\times MK}$ results from the K taps $\mathbf{X}_{\text{r}}(n-k,f) \in \mathbb{C}^{M\times M}$. The K taps $\mathbf{X}_{\text{r}}(n-k,f)$ are *dereverberated* versions of $x(n-k,f)$ obtained by applying the dereverberation filter $\underline{\mathbf{G}}(f)$ on $x(n-k,f)$:

$$\mathbf{X}_{\text{r}}(n-k,f) = x(n-k,f)\mathbf{I}_M - \sum_{l=\Delta}^{\Delta+L-1} x(n-k-l,f)\mathbf{G}(l,f). \tag{42}$$

$\mathbf{r}_d(n,f)$ is a *dereverberated* version of $\mathbf{d}(n,f)$ obtained by applying the dereverberation filter $\underline{\mathbf{G}}(f)$ on $\mathbf{d}(n,f)$ without prior echo cancellation:

$$\mathbf{r}_d(n,f) = \mathbf{d}(n,f) - \sum_{k=\Delta}^{\Delta+L-1} \mathbf{G}(k,f)\mathbf{d}(n-k,f) \tag{43}$$

Note that the update of the echo cancellation filter $\underline{\mathbf{H}}(f)$ is influenced by the dereverberation filter $\underline{\mathbf{G}}(f)$ through the terms $\underline{\mathbf{X}}_{\text{r}}(n,f)$ and $\mathbf{r}_d(n,f)$. This update prevents the echo cancellation filter $\underline{\mathbf{H}}(f)$ from reducing the component of the echo $\mathbf{y}(n,f)$ already reduced by the dereverberation filter $\underline{\mathbf{G}}(f)$.

The update of the echo cancellation filter $\underline{\mathbf{H}}(f)$ also depends on the PSDs $v_c(n,f)$ and the SCMs $\mathbf{R}_c(f)$ through the term $\mathbf{R}_{\mathbf{dd}}(n,f)$ defined in (35). As the post-filter $\mathbf{W}_c(n,f)$ is used for the updates of both of the PSDs $v_c(n,f)$ (see Section IV-B below) and the SCMs $\mathbf{R}_c(f)$ (see Section III-C4 below), the update of the echo cancellation filter $\underline{\mathbf{H}}(f)$ is also influenced by the post-filter $\mathbf{W}_c(n,f)$.

3) Dereverberation filter parameters Θ_G: Similarly to WPE for dereverberation [30], the dereverberation filter $\underline{\mathbf{G}}(f)$ is updated as

$$\underline{\mathbf{g}}(f) = \mathbf{Q}(f)^{-1}\mathbf{q}(f), \tag{44}$$

where

$$\mathbf{Q}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{E}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \underline{\mathbf{E}}(n,f), \tag{45}$$

$$\mathbf{q}(f) = \sum_{n=0}^{N-1} \underline{\mathbf{E}}(n,f)^H \mathbf{R}_{\mathbf{dd}}(n,f)^{-1} \mathbf{e}(n,f). \tag{46}$$

$\underline{\mathbf{g}}(f) = [\mathbf{g}_1(\Delta,f)^T \dots \mathbf{g}_M(\Delta,f)^T \dots\dots \mathbf{g}_1(\Delta+L-1,f)^T \dots \mathbf{g}_M(\Delta+L-1,f)^T]^T \in \mathbb{C}^{M^2L\times 1}$ is a vectorized version of $\underline{\mathbf{G}}(f)$, and $\underline{\mathbf{E}}(n,f) = [\mathbf{E}(n-\Delta,f) \dots \mathbf{E}(n-\Delta-L+1,f)] \in \mathbb{C}^{M\times M^2L}$ results from the L taps $\mathbf{E}(n-l,f) \in \mathbb{C}^{M\times M^2}$ obtained as

$$\mathbf{E}(n-l,f) = \mathbf{I}_M \otimes \mathbf{e}(n-l,f)^T. \tag{47}$$

The update of the dereverberation filter $\underline{\mathbf{G}}(f)$ is influenced by the echo cancellation filter $\underline{\mathbf{H}}(f)$ through the terms $\mathbf{e}(n,f)$. Similarly to the echo cancellation filter $\underline{\mathbf{H}}(f)$, the update of the dereverberation filter $\underline{\mathbf{G}}(f)$ is also influenced by the post-filter $\mathbf{W}_c(n,f)$ through the PSDs $v_c(n,f)$ and the SCMs $\mathbf{R}_c(f)$ used in the term $\mathbf{R}_{\mathbf{dd}}(n,f)$ defined in (35).

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, we estimate the variance and spatial covariance parameters 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

$$\mathbf{r}(n,f)\Big|\mathbf{d}(n-1,f),\ldots,\mathbf{d}(0,f),x(n,f),\ldots,x(0,f),\\ \underline{\mathbf{H}}(f),\underline{\mathbf{G}}(f)\sim\mathcal{N}_{\mathbb{C}}\Big(\mathbf{0},\mathbf{R}_{\mathbf{dd}}(n,f)\Big). \tag{48}$$

The signal model is conditionally identical to the local Gaussian modeling framework for source separation [37]. However, this framework does not constrain the PSDs or the SCMs which results in a permutation ambiguity (see Section II-C). 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 $\mathbf{b}_{\mathrm{r}}(n,f)$ [39]. In the E-step, each of these four sources $\mathbf{c}(n,f)$ is estimated as

$$\widehat{\mathbf{c}}(n,f)=\mathbf{W}_c(n,f)\mathbf{r}(n,f), \tag{49}$$

and its second-order posterior moment $\widehat{\mathbf{R}}_c(n,f)$ as

$$\widehat{\mathbf{R}}_c(n,f)=\widehat{\mathbf{c}}(n,f)\widehat{\mathbf{c}}(n,f)^H+\Big(\mathbf{I}-\mathbf{W}_c(n,f)\Big)v_c(n,f)\mathbf{R}_c(f). \tag{50}$$

In the M-step, we consider a weighted form of update for the SCMs $\mathbf{R}_c(f)$ [42]

$$\mathbf{R}_c(f)=\left(\sum_{n=0}^{N-1}w_c(n,f)\right)^{-1}\sum_{n=0}^{N-1}\frac{w_c(n,f)}{v_c(n,f)}\widehat{\mathbf{R}}_c(n,f), \tag{51}$$

where $w_c(n,f)$ denotes the weight of the source $\mathbf{c}(n,f)$. When $w_c(n,f)=1$, (51) reduces to the exact EM algorithm [37]. Here, we use $w_c(n,f)=v_c(n,f)$ [42], [43]. Experience shows that this weighting trick mitigates inaccurate estimates in certain time-frequency bins and increases the importance of the bins for which $v_c(n,f)$ is large. As the PSDs are constrained, we also need to constrain $\mathbf{R}_c(f)$ so as to encode only the spatial information of the sources. We modify (51) by normalizing $\mathbf{R}_c(f)$ after each update [42]:

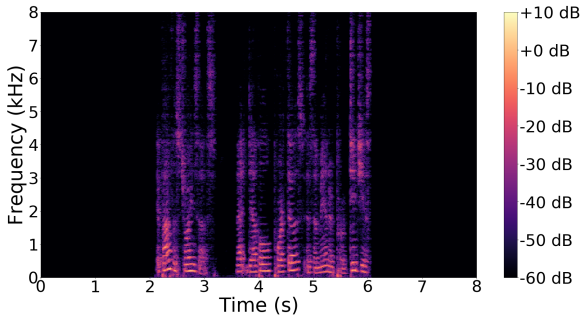
$$\mathbf{R}_c(f)\leftarrow\frac{M}{\mathrm{tr}\left(\mathbf{R}_c(f)\right)}\mathbf{R}_c(f). \tag{52}$$

The PSDs $v_c(n,f)$ of the four sources 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 IV below.

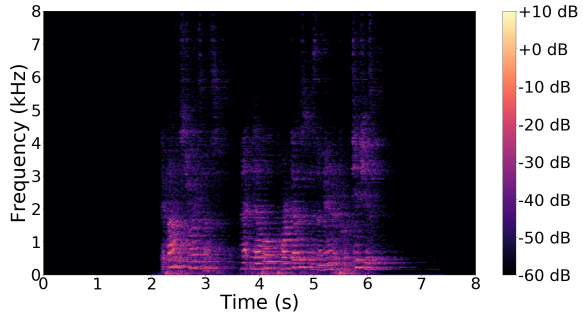
5) Estimation of the final early near-end component $\mathbf{s}_e(n,f)$*:* Once the proposed iterative optimization algorithm has converged after I iterations, we have estimates of the PSDs $v_c(n,f)$, the SCMs $\mathbf{R}_c(f)$ and the dereverberation filter $\underline{\mathbf{G}}(f)$. We can perform 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 (23), (24) and (49). For the detailed pseudo-code of the algorithm, please refer to the supporting document [41, Sec.3.5].

IV. NN SPECTRAL MODEL

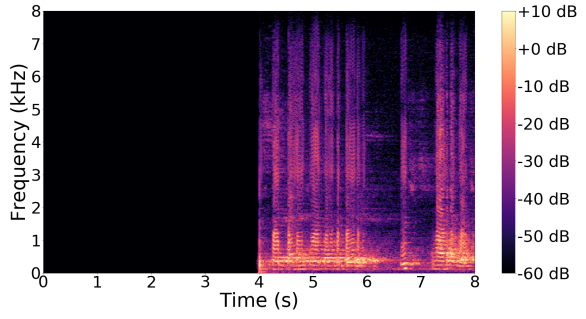
In this section, we define the inputs, the targets and the architecture of the NN used to initialize and update the target and residual PSDs.



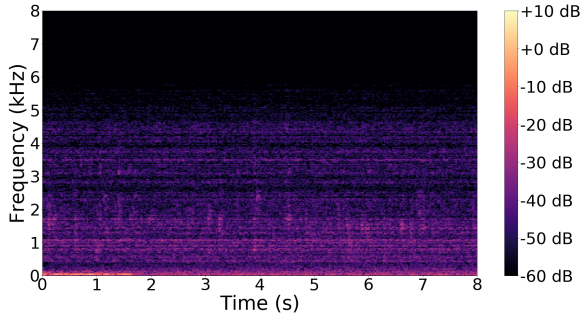
(a) Early near-end speech $\mathbf{s}_{\mathrm{e}}$



(b) Near-end residual late reverberation $\mathbf{s}_{\mathrm{r}}$



(c) *Dereverberated* residual echo $\mathbf{z}_{\mathrm{r}}$



(d) *Dereverberated* noise $\mathbf{b}_{\mathrm{r}}$

Fig. 5: Example ground truth target and residual signal PSDs in the training set.

A. 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 [39]. Therefore we define $\left[\sqrt{v_{s_{\mathrm{e}}}(n,f)}\sqrt{v_{s_{\mathrm{r}}}(n,f)}\sqrt{v_{z_{\mathrm{r}}}(n,f)}\sqrt{v_{b_{\mathrm{r}}}(n,f)}\right]$ as

the targets for the NN. Nugraha et al. defined the ground truth PSDs as $v_c(n,f) = \frac{1}{M}\|\mathbf{c}(n,f)\|^2$ [39]. We thus need to know the ground truth source signals $\mathbf{c}(n,f)$.

The ground truth latent signals $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ are unkown. However, in the training and validations sets, we can know the ground truth early near-end signal $\mathbf{s}_\mathrm{e}(n,f)$ and the signals $\mathbf{s}_\mathrm{l}(n,f)$, $\mathbf{y}(n,f)$ and $\mathbf{b}(n,f)$ (see Section V-B). These last three signals correspond to the values of $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ 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 procedure similar to the NN-supported BCA algorithm (see Fig. 4), where the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ are initialized to zero.

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. 4, respectively. We update $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$ by applying the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ to each of the signals $\mathbf{s}_\mathrm{l}(n,f)$, $\mathbf{y}(n,f)$ and $\mathbf{b}(n,f)$ as in (26), (27) and (28). To obtain the ground truth PSDs $v_c(n,f)$, we replace NN-EM at step 4 of Fig. 4 by an *oracle* estimation using Duong et al.'s EM algorithm [37]. For the detailed pseudo-code of the iterative procedure, please refer to the supporting document [41, Sec. 4.1]. After a few iterations, we observed the convergence of the latent variables $\mathbf{s}_\mathrm{r}(n,f)$, $\mathbf{z}_\mathrm{r}(n,f)$ and $\mathbf{b}_\mathrm{r}(n,f)$. In particular, we found that the *dereverberated* residual echo $\mathbf{z}_\mathrm{r}(n,f)$ decreases over the iterations. Fig. 5 shows an example of the PSD spectrograms after convergence.

B. 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 $\sqrt{v_c(n,f)}$ [39]. We concatenate these spectra to obtain the inputs. The different inputs are summarized in Fig. 6. 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 [42]

$$|\widetilde{d}(n,f)| = \sqrt{\frac{1}{M}\|\mathbf{d}(n,f)\|^2}. \tag{53}$$

Additionally we use the magnitude spectra $|\widetilde{y}(n,f)|$, $|\widetilde{e}(n,f)|$, $|\widetilde{e}_\mathrm{l}(n,f)|$ 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 [26]. We refer to the above inputs as type-I inputs. We consider additional inputs to improve the estimation. In particular, we use the magnitude spectra $\sqrt{v_c^{\mathrm{unc}}(n,f)}$ of the source unconstrained PSDs obtained as

$$v_c^{\mathrm{unc}}(n,f) = \frac{1}{M}\mathrm{tr}\left(\mathbf{R}_c(f)^{-1}\widehat{\mathbf{R}}_c(n,f)\right). \tag{54}$$

Indeed these inputs partially contain the spatial information of the sources and have been shown to improve results in source separation [39]. We refer to the inputs obtained from (54) as type-II inputs. For NN_0, we only use type-I inputs, as type-II inputs are not available at initialization. For NN_i with $i \geq 1$, we use both type-I and type-II inputs.

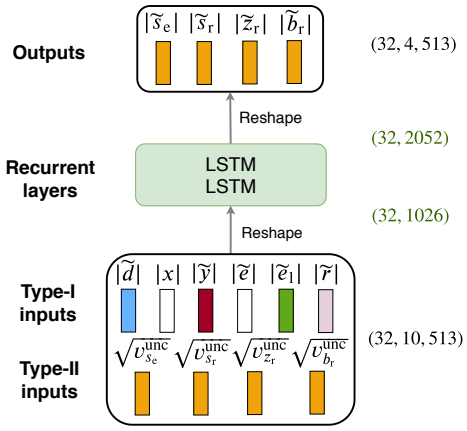


Fig. 6: Architecture of NN_i with a sequence length of 32 timesteps and $F = 513$ frequency bins.

C. Cost function

Let $|\widetilde{c}(n,f)|$ denote the NN output for source $\mathbf{c}(n,f)$. As mentioned above, we use NN_0 and NN_i to jointly predict the 4 spectral parameters $\left[|\widetilde{s}_\mathrm{e}(n,f)||\widetilde{s}_\mathrm{r}(n,f)||\widetilde{z}_\mathrm{r}(n,f)||\widetilde{b}_\mathrm{r}(n,f)|\right]$ (see Fig. 6). 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 [39]:

$$\mathcal{D}_{KL} = \frac{1}{4FN}\sum_{c,n,f}\Big(\sqrt{v_c(n,f)}\log\frac{\sqrt{v_c(n,f)}}{|\widetilde{c}(n,f)|} - \sqrt{v_c(n,f)} + |\widetilde{c}(n,f)|\Big). \tag{55}$$

D. Architecture

The neural network follows a long-short-term-memory (LSTM) network architecture. We consider 2 LSTM layers (see Fig. 6). 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.

V. Experimental protocol

In this section we describe the datasets, the metrics, the baselines and the hyperparameter settings used to evaluate the proposed algorithm.

A. Scenario

We consider the scenario where a near-end speaker interacts with a far-end speaker using a hands-free communication system at a distance of 1.5 m in a noisy environment. Each utterance has 8-s duration and contains 4 s of near-end speech and 4 s of far-end speech overlapping for 2 s. Background noise is present during the whole utterance. Each utterance is hence composed of 4 periods of 2 s as shown in Fig. 7: 1) noise only, 2) noise and near-end speech, 3) noise, near-end and far-end speech, 4) noise and far-end speech.

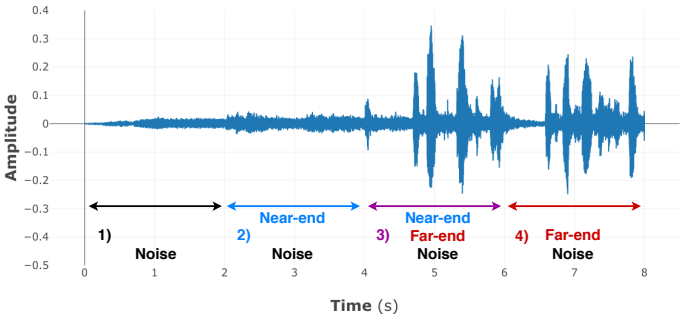


Fig. 7: Example utterance (only one channel shown).

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 I: Dataset characteristics.

B. Datasets

1) Overall description: We created three disjoint datasets for training, validation and test, whose characteristics are summarized in Table I. We considered $M = 3$ microphones. 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 (15). 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. The recording and simulation parameters (e.g. simulated room charateristics, position of the sources) are detailed in our companion technical report [41, Sec. 7.1].

a) Clean speech and noise signals: Clean speech signals were taken from the train-clean-360 subset of the Librispeech corpus [44], which consists of 921 speakers reading books for 25 min each on average. We selected 262 speakers and grouped them into 131 disjoint pairs for training, validation, and test. We alternately considered each speaker as near-end or far-end and picked several non-overlapping 4-s speech samples for each pair. Each 4-s sample was used only once in the whole dataset. Regarding the noise signals, we considered 6 types of domestic noise: babble, dishwasher, fridge, microwave, vacuum cleaner and washing machine. We randomly selected 78 non-overlapping 8-s noise samples from 1.7 h of YouTube videos and grouped them into disjoint subsets for training, validation, and test.

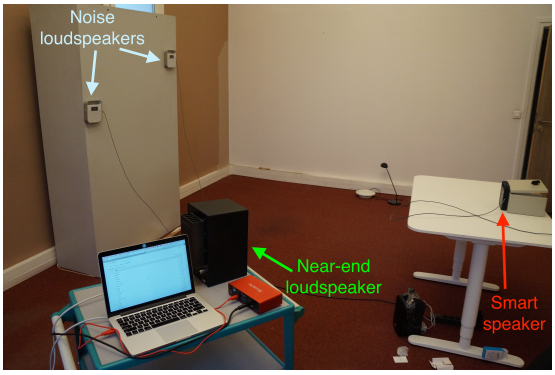


Fig. 8: Recording setup for the test set.

Room	Size (m)	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	$5.9 \times 4.6 \times 4$	1.3

TABLE II: Room characteristics.

b) Real echo recordings: To create the acoustic echo $\mathbf{y}(t)$, Togami et al. convolved far-end speech signals $x(t)$ with simulated echo paths $\mathbf{a}_y(\tau)$ which do not include any nonlinearity [19]. 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 (see Section II-A). 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. A configuration of the echo recording setup is given in Fig. 8. The recordings were done with the same Triby in 4 rooms with different size and reverberation time (RT_{60}) listed in Table II.

c) Reverberant 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.

2) Training set: For the training set, the echo recordings were done in rooms 1, 2 and 3 (see Table II). 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 to match the echo recording properties using the Roomsimove toolbox [45] [41, Sec. 7.1]. Among the 79 pairs of speakers used for training, 54 were used in rooms 1 and 2. We played and recorded 4,536 far-end signals and we simulated 4,536 near-end RIRs in each of these 2 rooms. The remaining 25 pairs were used in room 3. We played and recorded 4,500 far-end signals and we simulated 4,500 near-end RIRs in this room.

To create the noise signal $\mathbf{b}(t)$, we convolved a randomly chosen noise sample among the 36 noise samples (6 per noise type) used for training with the average of the late tail of two distinct RIRs randomly picked among 42 measured RIRs. This procedure approximates a spatially diffuse noise signal. To obtain the 42 measured RIRs, we measured 14 RIRs in

each of the rooms 1, 2 and 3.

The levels of the recorded far-end, the near-end speech and the noise signal were chosen randomly such that the signal-to-echo ratio (SER) varied from -45 dB to $+6$ dB and the signal-to-noise ratio (SNR) varied from -21 dB to $+24$ dB. These conditions are very challenging, especially as reverberation dominates in the reverberant near-end speech $\mathbf{s}(t)$. In total, we obtained 13,572 utterances which amount to roughly 32 h of audio.

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 similarly to the training set procedure. We played and recorded 4,536 far-end signals and we simulated 4,536 near-end RIRs in each room. To create the diffuse noise, we used the same 42 measured RIRs as in the training set. The levels of the recorded far-end speech, the near-end speech and the noise signal were chosen in the same range as the training set, resulting in the same challenging SER and SNR conditions. In total, we obtained 4,536 utterances which amount to roughly 10 h of audio.

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 II) using the setup shown in Fig. 8. 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. The noise signal $\mathbf{b}(t)$ was obtained by picking a random original noise signal and playing it through 4 Triby loudspeakers simutaneously. The noise signals resulting from this procedure are less diffuse than in the training and validation sets. The recorded levels were such that the resulting SER varied from -45 dB to -7 dB and the SNR varied from -20 dB to $+13$ dB. These challenging conditions are comprised within those of the training and validation sets. We played and recorded 4,500 far-end speech, near-end speech, and noise signals, hence we obtained a total of 4,500 8-s utterances amounting to 10 h of audio.

5) Time-varying test set: In order to evaluate our approach in time-varying acoustic conditions, we also considered the scenario when the near-end speaker speaks for 4 s, moves to a different position, and speaks for 4 s again. To do so, we concatenated pairs of 8-s near-end and echo recordings from the time-invariant test set corresponding to the same near-end and far-end speakers and microphone array positions, but to two different positions of the loudspeaker playing the near-end speech. The two recordings summed with an 16 s recorded noise signal. This resulted in 2,250 16-s utterances or roughly 10 h of audio.

C. Evaluation metrics

1) Early near-end components: The estimated early near-end signal $\widehat{\mathbf{s}}_{\mathrm{e}}(t)$ has 5 components

$$\widehat{\mathbf{s}}_{\mathrm{e}}(t) = \mathbf{s}_{\mathrm{e}}^{\mathrm{post}}(t) + \mathbf{s}_{\mathrm{l}}^{\mathrm{post}}(t) + \mathbf{y}^{\mathrm{post}}(t) + \mathbf{b}^{\mathrm{post}}(t) + \mathbf{s}_{\mathrm{e}}^{\mathrm{art}}(t), \quad (56)$$

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

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

where $\mathbf{s}_{\mathrm{e}}^{\mathrm{post}}(t)$ is the potentially attenuated early near-end signal, $\mathbf{s}_{\mathrm{l}}^{\mathrm{post}}(t)$, $\mathbf{y}^{\mathrm{post}}(t)$ and $\mathbf{b}^{\mathrm{post}}(t)$ are the post-residual distortion sources that are ideally equal to zero vectors, and $\mathbf{s}_{\mathrm{e}}^{\mathrm{art}}(t)$ denotes the artifacts introduced in the early near-end signal $\mathbf{s}_{\mathrm{e}}(t)$. This definition of the 5 components of the estimated target $\widehat{\mathbf{s}}_{\mathrm{e}}(t)$ is an extension of Le Roux et al.'s component definition in noise reduction to multiple distorsion sources [46]. For a detailed derivation of the components, please refer to the supporting document [41, Sec. 7.2]

2) Definition of the metrics: The objective metrics are summarized in Table III in the single channel case ($M = 1$). In the multichannel case ($M > 1$), we compute each metric on each channel m separately and we average the results over the M channels.

We evaluate the proposed joint approach in terms of the overall distortion, which is measured with the scale-invariant signal-to-distortion ratio (SI-SDR) [46]. The overall distortion takes the three distortion sources and the artifacts into account. To analyze the distribution of the overall distortion over the distortion sources and the artifacts, we use 5 additional metrics. For echo reduction, we use the SER and the echo return loss enhancement (ERLE) [3]. Dereverberation is assessed by the early-to-late reverberation ratio (ELR) [2]. We use this metric instead of the direct-to-reverberant ratio (DRR) [2] since early reflections are part of the target signal to be estimated. For noise reduction, we use the SNR. The artifacts are measured with the scale-invariant signal-to-artifacts ratio (SI-SAR) [46].

3) Evaluation period: The SI-SDR, ELR, SNR and SI-SAR are evaluated during both *single-talk* (near-end speech only) and *double-talk* (simultaneous near-end and far-end speech). The SER is only evaluated during *double-talk*, while the ERLE is evaluated during both *double-talk* and *far-end talk* (far-end speech only).

Since the performance may vary depending on the presence of acoustic echo which is the loudest signal, we compute the metrics separately for *near-end talk*, *double-talk* and *far-end*

talk. In particular, each metric depends on a scaling factor γ_c [41]. We assume that γ_c is constant during each *near-end talk*, *double-talk* or *far-end talk* period. However, γ_c may vary from one period to another. Finally, we average each metric over all periods in the fashion of the segmental SNR [47].

4) Ground truth signals: All the above metrics are based on the ground truth signals $\mathbf{s}_{\mathrm{e}}(t)$, $\mathbf{s}_{\mathrm{l}}(t)$, $\mathbf{y}(t)$ et $\mathbf{b}(t)$ (see Table III). The dataset generation procedure readily provides ground truth signals for the echo $\mathbf{y}(t)$ and the noise $\mathbf{b}(t)$. To define the ground truth signals of the target $\mathbf{s}_{\mathrm{e}}(t)$ and late reverberation $\mathbf{s}_{\mathrm{l}}(t)$, we set the mixing time as $t_{\mathrm{e}} = 64$ ms. We computed these two components using (8) which requires the ground truth near-end RIR $\mathbf{a}_s(\tau)$. In the test set, since the ground truth near-end RIR $\mathbf{a}_s(\tau)$ is unknown, we derive it using the evaluation procedure proposed by Yoshioka et al. for ELR[1] when $\mathbf{a}_s(\tau)$ is unknown [8, Sec. VII.A] [30, Sec. VI.A]. This evaluation procedure determines the ground truth $\mathbf{a}_s(\tau)$ by performing MMSE optimization between the reverberant near-end speech $\mathbf{s}(t)$ (output signal) and the anechoic near-end speech $u(t)$ (input signal).

D. 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 [19],
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 $\mathbf{W}_{s_{\mathrm{e}}}(n,f)$ are estimated and applied one after another. Echo cancellation relies on SpeexDSP[2], which implements Valin's adaptive approach and is particularly suitable for time-varying conditions [48] (see Section II-A). Dereverberation relies on our implementation of WPE [29], [30] (see Section II-B). The multichannel Wiener postfilter is computed using our implementation of Nugraha et al.'s NN-EM approach [39] (see Section II-C).
3) *NN-parallel*: a 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 Fig. 2),
4) *NN-cascade*: a variant of *Cascade* where the echo cancellation filter $\underline{\mathbf{H}}(f)$ is estimated using an NN-supported approach similar to *NN-joint* instead of Valin's adaptive approach. As WPE dereverberates similarly to its NN-supported counterpart in the multichannel case [32], *NN-cascade* corresponds to a cascade variant of *NN-joint* which estimates each filter separately using NN-supported optimization algorithms.

For the detailed description of the model and optimization algorithm of *NN-parallel* and *NN-cascade*, please refer to the supporting document [41, Sec. 5 and 6].

[1]The authors denoted this metric by DRR instead of ELR.

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

E. Hyperparameter settings

The hyperparameters of the three approaches are set as follows.

1) Initialization of the linear filters: For echo cancellation, we compute $\underline{\mathbf{H}}_0(f)$ by applying SpeexDSP on each channel of $\mathbf{d}(n,f)$. Since SpeexDSP relies on half-overlapping rectangular STFT windows, we use a window of length 512 and hopsize 256. We set the filter length to 0.208 s in the time domain, that is $K = 13$ frames. As SpeexDSP is an online algorithm, we apply it twice to each utterance to ensure convergence. For dereverberation, we compute $\underline{\mathbf{G}}_0(f)$ by performing 3 iterations of WPE on the signal $\mathbf{e}(t)$ output by SpeexDSP. We use the STFT with a Hanning window of length 1,024 and hopsize 256. We set the filter length to 0.208 s in the time domain, that is $L = 10$ frames, and the delay to $\Delta = 3$ frames.

2) Hyperparameters of the NNs: We consider 1026 units for the hidden layer of the LSTM structure. Regarding the activation functions, we use rectified linear units (ReLU) for the cell state of the layers and sigmoids for the gates. NN training is done by backpropagation with a minibatch size of 16 sequences, a fixed sequence length of 32 frames and the Adam parameter update algorithm with default settings [49]. To avoid gradient explosion with long sequences, we use gradient clipping with a threshold of 1.0. Training is stopped when the loss on the validation set stops decreasing for 5 epochs.

3) Hyperparameters of NN-joint: The STFT coefficients are computed with a Hanning window of length 1,024 and hopsize 256 resulting in $F = 513$ frequency bins. The length of the echo cancellation filter $\underline{\mathbf{H}}(f)$ (0.208 s in the time domain) now corresponds to $K = 10$ frames. The hyperparameters of the dereverberation filter $\underline{\mathbf{G}}(f)$ are identical to those of WPE. At training time, we perform 3 iterations of the iterative procedure to derive the ground truth PSDs (see Section IV-A) [41]. At test time, we perfom $I = 3$ iterations of the proposed NN-supported BCA algorithm with 1 spatial and 1 spectral update for each iteration i (see Fig. 4).

4) Hyperparameters of Togami : Togami requires the initial values for the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$, and for the PSDs of the reverberant near-end speech $v_s(n,f) = \frac{1}{M}\|\mathbf{s}(n,f)\|^2$ and the noise signal $v_b(n,f) = \frac{1}{M}\|\mathbf{b}(n,f)\|^2$. We initialize $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ by applying SpeexDSP and WPE on $\mathbf{d}(n,f)$, respectively, with the same hyperparameters as above. Since the authors did not specify how to initialize the PSDs [19], we estimate them using an NN similar to NN_0 where the type-I input for $|\widetilde{e}_{\mathrm{l}}(n,f)|$ is replaced by $|\widetilde{d}_{\mathrm{l}}(n,f)|$ obtained similarly to (53) 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. 2). All the SCMs are initialized to $\mathbf{I}_M$. We perform $I = 3$ iterations of *Togami*'s EM algorithm using the same STFT hyperparameters and values of K, L and Δ as for our approach.

5) Hyperparameters of Cascade: We compute and fix the linear filters to $\underline{\mathbf{H}}(f) = \underline{\mathbf{H}}_0(f)$ and $\underline{\mathbf{G}}(f) = \underline{\mathbf{G}}_0(f)$ with the same hyperparameters as *NN-joint*. Using $\underline{\mathbf{H}}_0(f)$ for echo cancellation is particularly efficient in time-varying conditions (see Section II-A). The NN architecture and inputs are identical to those in *NN-joint*, and the ground truth PSDs are

computed using the same procedure where the linear filters are fixed to $\underline{\mathbf{H}}(f) = \underline{\mathbf{H}}_0(f)$ and $\underline{\mathbf{G}}(f) = \underline{\mathbf{G}}_0(f)$ (see Section IV-A). Note that the type-I inputs for $|\widetilde{y}(n,f)|$, $|\widetilde{e}(n,f)|$, $|\widetilde{e}_1(n,f)|$ and $|\widetilde{r}(n,f)|$ remain fixed over the EM iterations because of the fixed linear filters.

6) Hyperparameters of NN-parallel: We compute the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ with the same hyperparameters as *NN-joint*. The NN architecture and inputs are identical to those in *NN-joint*, except for the type-I input for $|\widetilde{e}_1(n,f)|$ which is replaced by $|\widetilde{d}_1(n,f)|$ (see Section V-E4). The ground truth PSDs are computed using the same procedure as *NN-joint* but where the linear filters are applied in parallel [41]. We initialize $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ similarly to *Togami*.

7) Hyperparameters of NN-cascade: All the filters are computed with the same hyperparameters as *Cascade*. For echo cancellation, we compute $\underline{\mathbf{H}}_0(f)$ by applying the echo-only variant of *NN-joint*. To estimate $\underline{\mathbf{H}}_0(f)$, the NN architecture and inputs are identical to those in *NN-joint*, without the type-I inputs $|\widetilde{e}_1(n,f)|$ and $|\widetilde{r}(n,f)|$ related to dereverberation. We initialize the echo-only variant for estimating $\underline{\mathbf{H}}_0(f)$ by applying SpeexDSP on $\mathbf{d}(n,f)$ with the same hyperparameters as above. The ground truth PSDs of the echo-only variant for estimating $\underline{\mathbf{H}}_0(f)$ are computed using the same procedure as *NN-joint* without linear dereverberation. At test time, we perform $I = 3$ iterations of the echo-only variant for estimating $\underline{\mathbf{H}}_0(f)$ with 1 spatial and 1 spectral update for each iteration i.

8) Regularization: In order to avoid numerical instabilities and ill-conditioned matrices, we add a regularization scalar ϵ to the denominator in (51) and a regularization matrix $\epsilon\mathbf{I}$ to the matrix to be inverted in (31), (39) and (44). We also regularize the training loss in (55) similarly to Nugraha et al. [39]. We regularize likewise the four baseline approaches. The regularization hyperparameter is fixed to $\epsilon = 10^{-5}$.

VI. Results and Discussion

In this section, *NN-joint* is compared to *Togami*, *Cascade*, *NN-parallel* and *NN-cascade*. First, we investigate the influence of NN inputs on the performance of *NN-joint*. Secondly, we analyze the results of the five approaches in time-invariant conditions. Finally, we discuss their results in time-varying conditions and we compare their computation time. Audio examples are provided online[3].

A. Analysis of NN inputs

Fig. 9 shows the average SI-SDR of two NN input configurations of *NN-joint* 1) both type-I and type-II inputs are used, 2) only type-I inputs are used. In time-invariant conditions, configuration 1) outperforms configuration 2) in terms of SI-SDR starting from 2 iterations of the NN-supported BCA algorithm. This confirms that the type-II inputs improve the performance in source separation [39]. Note that for iteration $i = 0$, the two configurations are the same as type-II inputs are not available at initialization (see Fig. 6).

In time-varying conditions, the two configurations perform similarly in terms of SI-SDR, except for iteration $i = 1$.

[3]https://guillaumecarbajal.github.io/

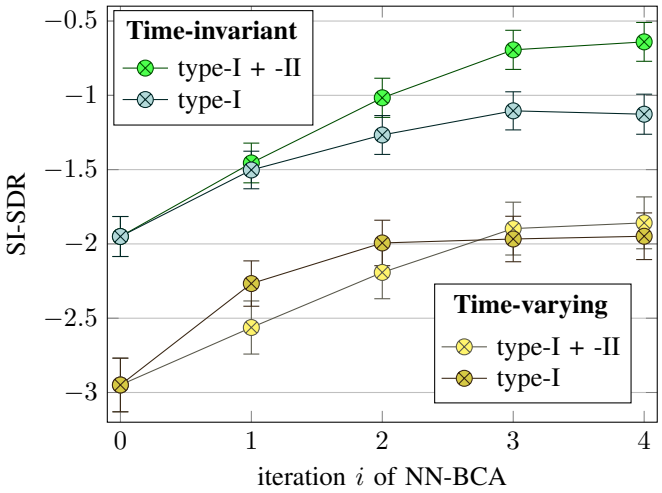


Fig. 9: Average overall distorsion results of *NN-joint* (in dB) w.r.t. NN inputs.

Indeed, the type-II inputs are computed with fixed SCMs $\mathbf{R}_c(f)$ while the spatial properties of target $\mathbf{s}_\text{e}$ and the near-end residual reverberation $\mathbf{s}_\text{r}$ vary over time. As a result, the type-II inputs do not improve NN estimation in configuration 1).

B. Time-invariant conditions

1) Average performance: Table IV shows the metrics related to the mixture $\mathbf{d}$. Fig. 10 shows the average results in time-invariant conditions. All approaches have a negative SI-SDR, which is caused by the challenging test set conditions.

NN-joint outperforms *Togami* by 3.8 dB in terms of SI-SDR. *NN-parallel* provides information about this difference in performance since it applies the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ in the same order as in *Togami*, but uses a similar signal model and optimization algorithm as *NN-joint* (see Section II-D). *NN-parallel* also outperforms *Togami* by 3.8 dB in terms of SI-SDR. Therefore our proposed signal model and optimization algorithm explain the SI-SDR difference with *Togami*. Although the parallel variant achieves lower reduction of echo, reverberation and noise than *Togami*, it introduces lower degradation in the target $\mathbf{s}_\text{e}$. Regarding *NN-joint* and *NN-parallel*, applying the linear filters $\underline{\mathbf{H}}(f)$ and $\underline{\mathbf{G}}(f)$ one after another only modifies the distribution of the overall distortion over the echo (greater reduction), reverberation (greater reduction) and noise (lower reduction).

NN-joint outperforms *Cascade* 1.0 dB in terms of SI-SDR. *NN-cascade* provides information about this difference in

SI-SDR	ERLE	SER	ELR	SNR
$\mathbf{-16.2 \pm 0.2}$	0.0	-25.8 ± 0.3	-1.1 ± 0.1	-2.6 ± 0.2

TABLE IV: Metrics (in dB) related to the mixture signal $\mathbf{d}$ in the test set. The metrics are computed using the decomposition in (15). ERLE $= 0$ dB since there is no echo reduction. The SI-SAR is not computed since there is no artifacts in the unprocessed target $\mathbf{s}_\text{e}$.

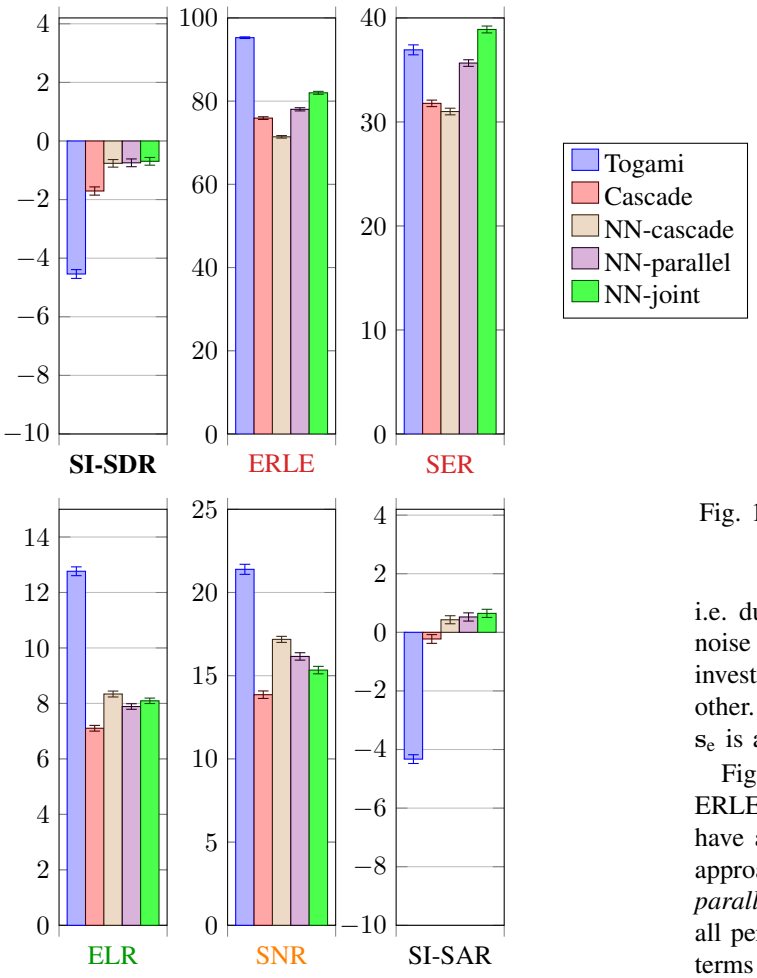


Fig. 10: Average results (in dB) in time-invariant conditions

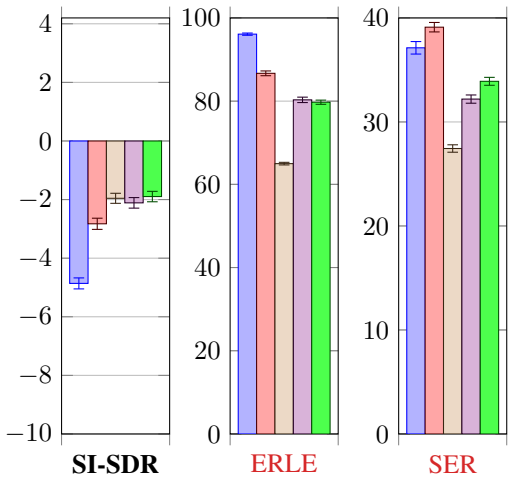


Fig. 11: Average results (in dB) in time-varying conditions.

performance since it also uses an NN-supported echo cancellation as in *NN-joint*, but estimates each filter separately. *NN-cascade* also outperforms *Cascade* by 1.0 dB in terms of SI-SDR. Therefore the proposed NN-supported echo cancellation in *NN-joint* explains the SI-SDR difference with *Cascade*. Regarding the optimization of the filters, jointly optimizing them modifies the distribution of the overall distortion over the echo (greater reduction), reverberation (lower reduction) and noise (lower reduction).

From informal listening tests, we can state that the estimated target speech $\widehat{\mathbf{s}}_\mathrm{e}$ is often highly attenuated and distorted during double-talk for all approaches when SER ≤ -20 dB. Regarding *Togami*, a slight reverberation remains, but noise and echo seem completely removed. However, the estimated target speech $\widehat{\mathbf{s}}_\mathrm{e}$ is much more attenuated and distorted than with the other approaches, especially during *double-talk* where the estimated target speech $\widehat{\mathbf{s}}_\mathrm{e}$ can never be heard. Regarding the other approaches, post-residual distortions seem louder than with *Togami*, but a comparison between these approaches is difficult to make.

2) Interactions of system components: While the above results show 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 simultaneously, 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 $\mathbf{s}_\mathrm{e}$ is absent in this scenario.

Fig. 12 shows the results during *near-end talk*. SER and ERLE are not evaluated as echo is absent. All approaches have a positive SI-SDR. The SI-SAR is also positive for all approaches. The trend in performance between *NN-joint*, *NN-parallel* and *Togami* is similar to the results averaged over all periods. *NN-cascade* outperforms *Cascade* by $+0.6$ dB in terms of SI-SDR. This is due to greater dereverberation and noise reduction with comparable degradation of the target $\mathbf{s}_\mathrm{e}$. This might be due to the performance before post-filtering [41, Sec. 8.1]: linear dereverberation in *NN-cascade* also achieves greater dereverberation and noise reduction than *Cascade*. This results from echo cancellation: since the dereverberation filter $\underline{\mathbf{G}}(f)$ is time-invariant, its performance during *near-end talk* is also affected by echo cancellation during *double-talk*. In *NN-cascade*, the NN-supported echo cancellation achieves greater echo reduction than Valin's echo cancellation in *Cascade*. As a result, linear dereverberation in *NN-cascade* is able to achieve greater reduction of the other distortion signals, i.e. reverberation and noise.

NN-cascade also outperforms *NN-joint* in terms of SI-SDR during *near-end talk*. Indeed, joint estimation of the filters in *NN-joint* implies a performance compromise during all periods in order to reduce all the distortion signals. In the case of *NN-cascade*, there is no performance compromise as the filters are estimated separately. Thus *NN-cascade* might better perform when one distortion source is absent. Hence the greater performance of *NN-cascade* during *near-end talk* where echo is absent. All in all, *NN-joint* does not improve performance when only reverberation and noise are present, but does not degrade it either compared to *Cascade*.

Fig. 13 shows the results during *double-talk*. The trend in performance between *NN-joint*, *NN-parallel* and *Togami* is similar to the results averaged over all periods. *NN-cascade* outperforms *Cascade* by 1.2 dB in terms of SI-SDR. *NN-*

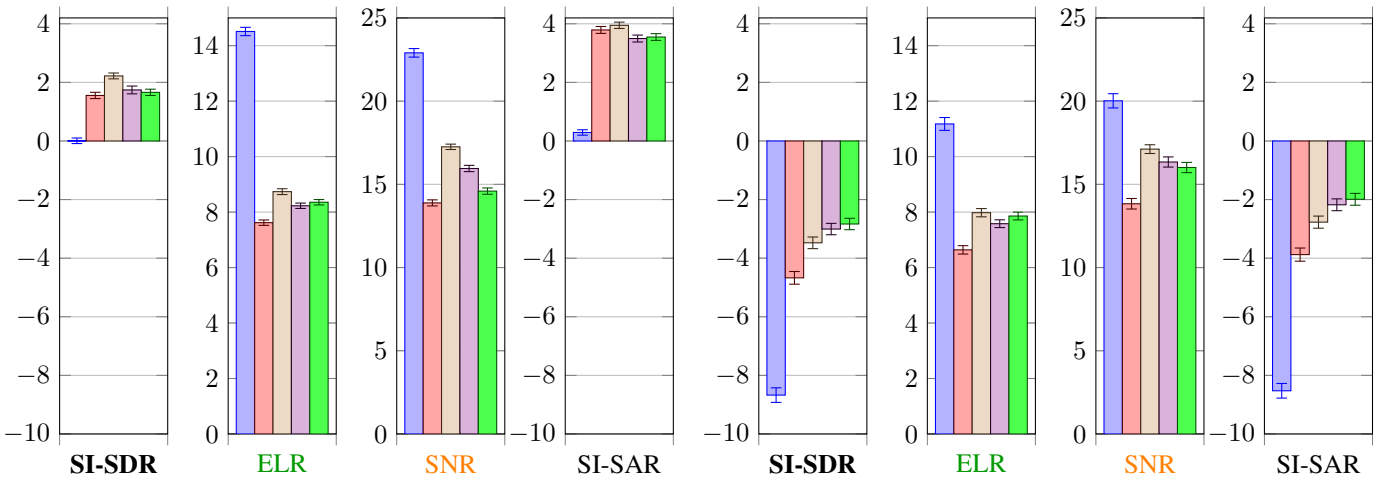


Fig. 12: Results (in dB) during near-end talk in time-invariant conditions.

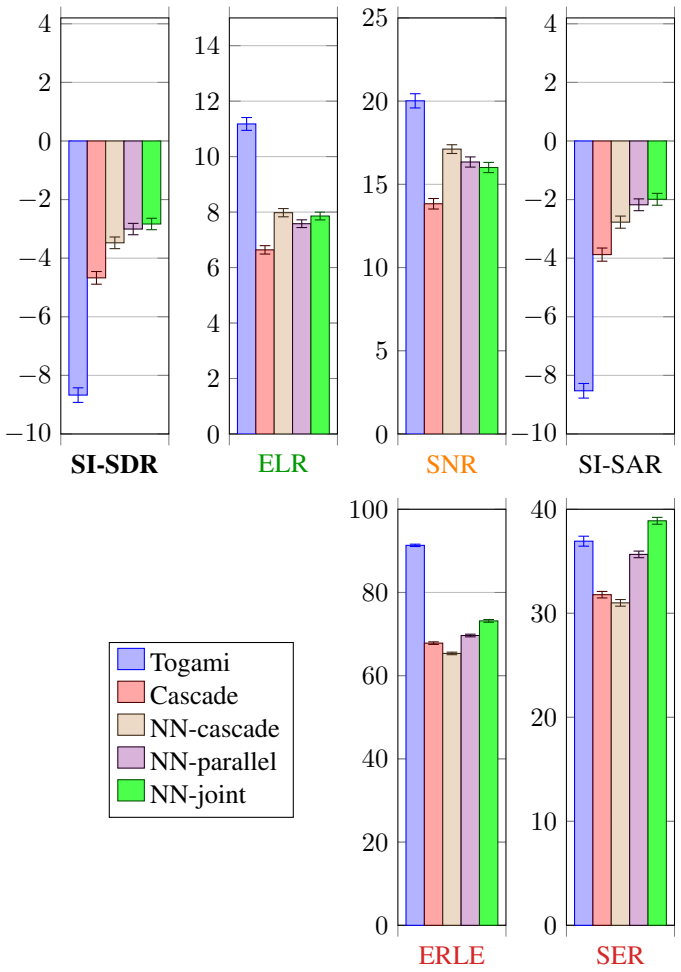


Fig. 13: Results (in dB) during double-talk in time-invariant conditions.

joint outperforms *NN-cascade* by 0.6 dB in terms of SI-SDR. Therefore, during *double-talk*, both joint optimization of the filters and the proposed NN-supported echo cancellation in *NN-joint* explain the SI-SDR improvement between *NN-joint* and *Cascade*. Although *NN-joint* achieves lower dereverberation and noise reduction than *NN-cascade*, it achieves greater echo reduction with lower degradation of the target $\mathbf{s}_\mathrm{e}$. As a result, *NN-joint* improves performance when echo, reverberation and noise are present simultaneously.

From *near-end talk* to *double-talk*, the SI-SDR decreases by 4.5 dB for *NN-joint*, by 5.7 dB for *NN-cascade* and by 6.2 dB for *Cascade*. We conclude that *NN-joint* improves robustness in terms of SI-SDR when echo, reverberation and noise are present simultaneously, while not degrading performance when only reverberation and noise are present.

C. Time-varying conditions

Fig. 11 shows the average results in time-varying conditions. As the trend in ELR, SNR and SAR is similar to the average results in time-invariant conditions for all approaches, we discard these metrics from the analysis and we provide them in the supporting document [41, Sec. 8.2]. The SI-SDR is lower than in time-invariant conditions for all approaches (see Fig. 10) since the spatial properties of target $\mathbf{s}_\mathrm{e}$ and the near-end residual reverberation $\mathbf{s}_\mathrm{r}$ vary over time while their SCMs $\mathbf{R}_c(f)$ remain fixed. This also explains the drop in SI-SDR for the two NN input configurations of *NN-joint* in time-varying conditions (see Fig. 9). Informal listening tests provide the same observations as in time-invariant 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 also similar to the average SI-SDR in time-invariant conditions. However, *Cascade* achieves here the greatest echo reduction. *Cascade* also systematically achieves the greatest echo reduction after echo cancellation, and also after echo cancellation and dereveberation [41, Sec. 8.1]. This is explained by Valin's adaptive approach for echo cancellation which is designed for time-varying conditions [48] (see Section II-A).

Togami	NN-cascade	NN-parallel	NN-joint
$-\mathbf{42}\%$	$+56\%$	$+7\%$	$+16\%$

TABLE V: Computation time of the approaches compared to *Cascade* (in percentage).

D. Computation time

We discard the initialization as it is the same for all 5 approaches (see Section V-E). We compute the target $\widehat{\mathbf{s}}_\mathrm{e}$ for a 8 s utterance with a 2.7 GHz Intel Core i5 CPU. Table V shows the computation time of the approaches compared to the cascade approach. *NN-joint* is much faster than *NN-cascade*. Therefore joint optimization of the filters significantly reduces the computation time. In addition, *NN-parallel* is slightly faster than *NN-joint*. Since *Cascade* is one of the approaches implemented in today's industrial devices, we conclude that both *NN-joint* and *NN-parallel* could be implemented in real time.

VII. Conclusion

We proposed an NN-supported BCA algorithm for joint multichannel reduction of acoustic echo, reverberation and noise. The approach jointly models the spectra of the target and residual signals after echo cancellation and dereverberation with an NN. We evaluated our system on real recordings of acoustic echo, reverberation and noise acquired with a smart speaker in various situations. When echo, reverberation and noise are present simultaneously, the proposed approach outperforms the cascade approach and Togami et al.'s joint reduction approach in terms of overall distortion reduction while not degrading performance when only reverberation and noise are present. Future work will focus on a recursive version of the approach in order to better handle time-varying conditions.

References

[1] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[2] P. A. Naylor and N. D. Gaubitch, Eds., *Speech Dereverberation*. Springer, 2010.

[3] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: a Practical Approach*. Wiley-Interscience, 2004.

[4] J. S. Erkelens and R. Heusdens, "Correlation-based and model-based blind single-channel late-reverberation suppression in noisy time-varying acoustical environments," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1746–1765, 2010.

[5] I. Kodrasi and S. Doclo, "Joint dereverberation and noise reduction based on acoustic multi-channel equalization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 680–693, 2016.

[6] O. Schwartz, S. Gannot, and E. A. P. Habets, "Multi-microphone speech dereverberation and noise reduction using relative early transfer functions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 240–251, 2015.

[7] T. Dietzen, S. Doclo, M. Moonen, and T. van Waterschoot, "Joint multi-microphone speech dereverberation and noise reduction using integrated sidelobe cancellation and linear prediction," in *Proc. IWAENC*, 2018, pp. 221–225.

[8] 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.

[9] H. Kagami, H. Kameoka, and M. Yukawa, "Joint separation and dereverberation of reverberant mixtures with determined multichannel non-negative matrix factorization," in *Proc. ICASSP*, 2018, pp. 31–35.

[10] R. Le Bouquin Jeannès, P. Scalart, G. Faucon, and C. Beaugeant, "Combined noise and echo reduction in hands-free systems: a survey," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 808–820, 2001.

[11] S. Gustafsson, R. Martin, P. Jax, and P. Vary, "A psychoacoustic approach to combined acoustic echo cancellation and noise reduction," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 245–256, 2002.

[12] W. Herbordt, S. Nakamura, and W. Kellermann, "Joint optimization of LCMV beamforming and acoustic echo cancellation for automatic speech recognition," in *Proc. ICASSP*, 2005, pp. III–77–80.

[13] G. Reuven, S. Gannot, and I. Cohen, "Joint noise reduction and acoustic echo cancellation using the transfer-function generalized sidelobe canceller," *Speech Communication*, vol. 49, no. 7–8, pp. 623–635, 2007.

[14] M. Togami, Y. Kawaguchi, and R. Takashima, "Frequency domain acoustic echo reduction based on Kalman smoother with time-varying noise covariance matrix," in *Proc. ICASSP*, 2014, pp. 5909–5913.

[15] K. Nathwani, "Joint acoustic echo and noise cancellation using spectral domain Kalman filtering in double talk scenario," in *Proc. IWAENC*, 2018, pp. 326–330.

[16] R. Takeda, K. Nakadai, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno, "ICA-based efficient blind dereverberation and echo cancellation method for barge-in-able robot audition," in *Proc. ICASSP*, 2009, pp. 3677–3680.

[17] M. Togami and Y. Kawaguchi, "Speech enhancement combined with dereverberation and acoustic echo reduction for time varying systems," in *Proc. SSP*, 2012, pp. 357–360.

[18] E. A. P. Habets, S. Gannot, I. Cohen, and P. C. Sommen, "Joint dereverberation and residual echo suppression of speech signals in noisy environments," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1433–1451, 2008.

[19] 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.

[20] D. S. Williamson and D. Wang, "Speech dereverberation and denoising using complex ratio masks," in *Proc. ICASSP*, 2017, pp. 5590–5594.

[21] Y. Zhao, Z.-Q. Wang, and D. Wang, "A two-stage algorithm for noisy and reverberant speech enhancement," in *Proc. ICASSP*, 2017, pp. 5580–5584.

[22] H. Seo, M. Lee, and J.-H. Chang, "Integrated acoustic echo and background noise suppression based on stacked deep neural networks," *Applied Acoustics*, vol. 133, pp. 194–201, 2018.

[23] H. Zhang and D. Wang, "Deep learning for acoustic echo cancellation in noisy and double-talk scenarios," in *Interspeech*, 2018, pp. 3239–3243.

[24] F. Yang, G. Enzner, and J. Yang, "Statistical convergence analysis for optimal control of DFT-domain adaptive echo canceler," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1095–1106, 2017.

[25] C. M. Lee, J. W. Shin, and N. S. Kim, "DNN-based residual echo suppression," in *Interspeech*, 2015, pp. 316–320.

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

[27] G. Enzner and P. Vary, "Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones," *Signal Processing*, vol. 86, no. 6, pp. 1140–1156, 2006.

[28] M. Togami and K. Hori, "Multichannel semi-blind source separation via local Gaussian modeling for acoustic echo reduction," in *Proc. EUSIPCO*, 2011, pp. 496–500.

[29] 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.

[30] 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.

[31] A. Jukic, T. van Waterschoot, T. Gerkmann, and S. Doclo, "Multi-Channel Linear Prediction-Based Speech Dereverberation With Sparse Priors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1509–1520, 2015.

[32] 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.

[33] K. Furuya and A. Kataoka, "Robust speech dereverberation using multichannel blind deconvolution with spectral subtraction," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 5, pp. 1579–1591, 2007.

[34] M. Togami, Y. Kawaguchi, R. Takeda, Y. Obuchi, and N. Nukaga, "Optimized speech dereverberation from probabilistic perspective for time varying acoustic transfer function," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1369–1380, 2013.

[35] A. Cohen, G. Stemmer, S. Ingalsuo, and S. Markovich-Golan, "Combined weighted prediction error and minimum variance distortionless response for dereverberation," in *Proc. ICASSP*, 2017, pp. 446–450.

[36] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[37] 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.

[38] 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.

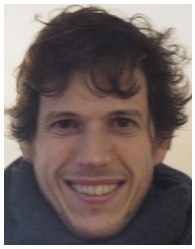
[39] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on*

Audio, Speech, and Language Processing, vol. 24, no. 9, pp. 1652–1664, 2016.
[40] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *Proc. ICASSP*, 2019, pp. 101–105.
[41] 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. [Online]. Available: https://hal.inria.fr/hal-02372431
[42] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. EUSIPCO*, 2016, pp. 1748–1752.
[43] A. Liutkus, D. Fitzgerald, and Z. Rafii, "Scalable audio separation with light kernel additive modelling," in *Proc. ICASSP*, 2015, pp. 76–80.
[44] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.
[45] E. Vincent and D. R. Campbell, "Roomsimove," 2008. [Online]. Available: http://homepages.loria.fr/evincent/software/Roomsimove_1.4.zip
[46] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR — half-baked or well done?" in *Proc. ICASSP*, 2019, pp. 626–630.
[47] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2007.
[48] 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.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

Guillaume Carbajal received the M.Sc. degree in machine learning from the Université Paris-Saclay (Paris, France) in 2016 and the State Engineering from Ecole Nationale des Ponts et Chaussées (Paris, France) in 2017. He worked as an audio research engineer at Invoxia (Issy-les-Moulineaux, France) from 2017 to 2020 and obtained the Ph.D. degree in informatics from the Université de Lorraine and Inria Nancy–Grand-Est (Nancy, France) in 2020. He is currently a postdoctoral researcher at the computer science department of the Universität Hamburg (Hamburg, Germany). His research interests include speech enhancement, audio-visual signal processing and machine learning.

Romain Serizel received the M.Eng. degree in Automatic System Engineering from ENSEM (Nancy, France) in 2005 and the M.Sc. degree in Signal Processing from Université Rennes 1 (Rennes, France) in 2006. He received the Ph.D. degree in Engineering Sciences from the KU Leuven (Leuven, Belgium) in June 2011. From 2011 till 2016 he was a postdoctoral researcher at KU Leuven (Leuven, Belgium), FBK (Trento, Italy), and at Télécom ParisTech (Paris, France). He is now an Associate Professor with Université de Lorraine (Nancy, France) doing research on robust speech communications and ambient sound detection and classification. Since 2019, he is coordinating the DCASE challenge series together with Annamaria Mesaros.

Emmanuel Vincent is a Senior Research Scientist with Inria (Nancy, France). He received the Ph.D. degree in music signal processing from the Institut de Recherche et Coordination Acoustique/Musique (Ircam, Paris, France) in 2004 and worked as a Research Assistant with the Centre for Digital Music at Queen Mary, University of London (United Kingdom), from 2004 to 2006. His research focuses on statistical machine learning for speech and audio signal processing, with application to audio source localization and separation, speech enhancement, and robust speech and speaker recognition. He is a founder of the series of CHiME, SiSEC and VoicePrivacy challenges. He was an associate editor for IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.



Eric Humbert received the State Engineering from Centrale-Supélec (Paris, France) in 2002 and the M.Sc. degree in acoustics, computer science and signal processing applied to music (ATIAM) from the Université Pierre et Marie Curie (Paris, France) in 2003. He worked as an audio software engineer at Inventel (Paris, France) from 2003 to 2005, at Thomson (Paris, France) from 2005 to 2009 and at Technicolor from 2009 to 2010. He is currently the AI director at Invoxia (Paris, France).