



**HAL**  
open science

# Multichannel audio source separation with deep neural networks

Aditya Arie Nugraha, Antoine Liutkus, Emmanuel Vincent

► **To cite this version:**

Aditya Arie Nugraha, Antoine Liutkus, Emmanuel Vincent. Multichannel audio source separation with deep neural networks. [Research Report] RR-8740, Inria. 2016. hal-01163369v3

**HAL Id: hal-01163369**

**<https://inria.hal.science/hal-01163369v3>**

Submitted on 5 Feb 2016 (v3), last revised 21 Jun 2016 (v5)

**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.



# Multichannel audio source separation with deep neural networks

Aditya Arie Nugraha, Antoine Liutkus, Emmanuel Vincent

**RESEARCH  
REPORT**

**N° 8740**

June 2015

Project-Team MULTISPEECH





## Multichannel audio source separation with deep neural networks

Aditya Arie Nugraha, Antoine Liutkus, Emmanuel Vincent

Project-Team MULTISPEECH

Research Report n° 8740 — version 2 — initial version June 2015 —  
revised version January 2016 — 25 pages

**Abstract:** This research report addresses the problem of multichannel audio source separation. We propose a deep neural network (DNN) based framework where the source spectra are estimated using DNNs and used in a multichannel filter. The filter is derived using an iterative expectation-maximization (EM) algorithm, in which spatial covariance matrices encode the spatial information. We present an extensive experimental study to show the impact of different design choices on the performance of the proposed technique. We consider different cost functions for the training of DNNs, namely Itakura-Saito (IS) divergence, Cauchy cost function, phase-sensitive cost function, and mean squared error (MSE). The use of probabilistically motivated cost function, such as the IS divergence, is interesting because it leads to a mathematically rigorous EM interpretation for the proposed framework. We also study the number of EM iterations and the use of multiple DNNs, where each DNN aims to improve the spectra estimated by the preceding EM iteration. Finally, we present its application to a speech enhancement problem. The experimental results show the benefit of the proposed multichannel approach over a single-channel DNN-based approach.

**Key-words:** audio source separation, speech enhancement, multichannel, deep neural networks (DNN), expectation-maximization (EM)

**RESEARCH CENTRE  
NANCY – GRAND EST**

615 rue du Jardin Botanique  
CS20101  
54603 Villers-lès-Nancy Cedex

# Séparation de sources audio multicanale par réseaux de neurones profonds

**Résumé :** Ce rapport porte sur le problème de la séparation de sources audio multicanal. Nous proposons un cadre basé sur les réseaux de neurones profonds (*deep neural networks* ou DNNs) où les spectres des sources sont estimés par des DNNs et utilisés dans un filtre multicanal. Ce filtre est obtenu en utilisant un algorithme d'espérance-maximisation (EM), dans lequel des matrices de covariance spatiale encodent l'information spatiale. Nous présentons une étude expérimentale approfondie montrant l'impact de différents choix sur la performance de la technique proposée. Nous considérons différentes fonctions de coût pour l'apprentissage des DNNs: la divergence d'Itakura-Saito (IS) divergence, la fonction de coût de Cauchy, une fonction de coût sensible à la phase, et l'erreur quadratique moyenne (*mean squared error* ou MSE). L'usage de fonctions de coût motivées par la théorie des probabilités, comme la divergence IS et la fonction de coût de Cauchy, est intéressant car il permet une interprétation probabiliste rigoureuse de l'algorithme EM proposé. Nous étudions aussi le nombre d'itérations EM et l'usage de DNNs multiples, où chaque DNN vise à améliorer les spectres estimés à l'itération EM précédente. Enfin, nous présentons l'application de cette technique à un problème de rehaussement de la parole. Les résultats expérimentaux montrent le potentiel de l'approche multicanal proposée par rapport à une approche basée sur les DNNs à un seul canal.

**Mots-clés :** séparation de sources audio, rehaussement de la parole, multicanal, réseaux de neurones profonds, algorithme d'Espérance-Maximisation (EM)

## 1 Introduction

Audio source separation aims to recover the signals of underlying sound sources from an observed mixture signal. Recent research on source separation can be divided into (1) speech separation, in which the speech signal is recovered from a mixture containing multiple background noise sources with possibly interfering speech; and (2) music separation, in which the singing voice and possibly certain instruments are recovered from a mixture containing multiple musical instruments. Speech separation is mainly used for speech enhancement in hearing aids or noise robust automatic speech recognition (ASR), while music separation has many interesting applications, including music editing/remixing, upmixing, music information retrieval, and karaoke [1–5].

Recent studies have shown that deep neural networks (DNNs) are able to model complex functions and perform well on various tasks, notably ASR [6, 7]. More recently, DNNs have been applied to single-channel speech enhancement and shown to provide a significant increase in ASR performance compared to earlier approaches based on beamforming or nonnegative matrix factorization (NMF) [8]. The DNNs typically operate on magnitude or log-magnitude spectra in the Mel domain or the short time Fourier transform (STFT) domain. Various other features have been studied in [9–11]. The DNNs can be used either to predict the source spectrograms [11–16] whose ratio yields a time-frequency mask or directly to predict a time-frequency mask [10, 17–21]. The estimated source signal is then obtained as the product of the input mixture signal and the estimated time-frequency mask. Various DNN architectures and training criteria have been investigated and compared [19, 21, 22]. Although the authors in [15] considered both speech and music separation, most studies focused either on speech separation [10–12, 14, 17–21] or on music separation [13, 16].

As shown in many works mentioned above, the use of DNNs for audio source separation by modeling the spectral information is extremely promising. However, a framework to exploit DNNs for multichannel audio source separation is lacking. Most of the approaches above considered single-channel source separation, where the input signal is either one of the channels of the original multichannel mixture signal or the result of delay-and-sum (DS) beamforming [19]. Efforts on exploiting multichannel data have been done by extracting multichannel features and using them to derive a single-channel mask [10, 11]. As a result, they do not fully exploit the benefits of multichannel data as achieved by multichannel filtering [1, 4].

In this research report, we propose a DNN-based multichannel source separation framework where the source spectra are estimated using DNNs and used to derive a multichannel filter using an iterative EM algorithm. The framework is built upon the state-of-the-art iterative EM algorithm in [23] which integrates spatial and spectral models in a probabilistic fashion. This model was used up to some variants in [24–28]. We study the impact of different design choices on the performance, including the cost function used for the training of DNNs and the number of EM iterations. We also study the impact of the spatial information by varying the number of spatial parameter updates and the use of multiple DNNs to improve the spectra over the iterations. We present the application of the proposed framework to a speech enhancement problem.

This work extends our preliminary work in [29] by following the exact EM algorithm in [24], instead of its variant in [28] and by reporting extensive experiments to study the impact of different design choices not only on the speech recognition performance, but also on the source separation performance.

The rest of this research report is organized as follows. Section 2 describes the iterative EM algorithm for multichannel source separation, which is the basis for the proposed DNN-based iterative algorithm described in Section 3. Section 4 shows the effectiveness of the framework for a speech separation problem and the impact of different design choices. Finally, Section 5

concludes the research report and presents future directions.

## 2 Background

In this section, we briefly describe the problem of multichannel source separation and the iterative EM algorithm in [23, 24], which is the basis for the proposed DNN-based multichannel source separation algorithm.

### 2.1 Problem formulation

Following classical source separation terminology [5], let  $I$  denote the number of channels,  $J$  the number of sources,  $\mathbf{c}_j(t) \in \mathbb{R}^{I \times 1}$  the  $I$ -channel spatial image of source  $j$ , and  $\mathbf{x}(t) \in \mathbb{R}^{I \times 1}$  the observed  $I$ -channel mixture signal. Both  $\mathbf{c}_j(t)$  and  $\mathbf{x}(t)$  are in the time domain and related by

$$\mathbf{x}(t) = \sum_{j=1}^J \mathbf{c}_j(t). \quad (1)$$

Source separation aims to recover the source spatial images  $\mathbf{c}_j(t)$  from the observed mixture signal  $\mathbf{x}(t)$ .

### 2.2 Model

Let  $\mathbf{x}(f, n) \in \mathbb{C}^{I \times 1}$  and  $\mathbf{c}_j(f, n) \in \mathbb{C}^{I \times 1}$  denote the short-time Fourier transform (STFT) coefficients of  $\mathbf{x}(t)$  and  $\mathbf{c}_j(t)$ , respectively, for frequency bin  $f$  and time frame  $n$ . Also, let  $F$  be the number of frequency bins and  $N$  the number of time frames.

We assume that  $\mathbf{c}_j(f, n)$  are independent of each other and follow a multivariate complex-valued zero-mean Gaussian distribution [23, 24, 27, 30]

$$\mathbf{c}_j(f, n) \sim \mathcal{N}_c(\mathbf{0}, v_j(f, n)\mathbf{R}_j(f)), \quad (2)$$

where  $v_j(f, n) \in \mathbb{R}_+$  denotes the power spectral density (PSD) of source  $j$  for frequency bin  $f$  and time frame  $n$ , and  $\mathbf{R}_j(f) \in \mathbb{C}^{I \times I}$  is the spatial covariance matrix of source  $j$  for frequency bin  $f$ . This  $I \times I$  matrix represents spatial information by encoding the spatial position and the spatial width of the corresponding source [23]. Since the mixture  $\mathbf{x}(f, n)$  is the sum of  $\mathbf{c}_j(f, n)$ , it is consequently distributed as

$$\mathbf{x}(f, n) \sim \mathcal{N}_c\left(\mathbf{0}, \sum_{j=1}^J v_j(f, n)\mathbf{R}_j(f)\right). \quad (3)$$

Given the PSDs  $v_j(f, n)$  and the spatial covariance matrices  $\mathbf{R}_j(f)$  of all sources, the spatial source images can be estimated in the minimum mean square error (MMSE) sense using multichannel Wiener filtering [23, 27]

$$\hat{\mathbf{c}}_j(f, n) = \mathbf{W}_j(f, n)\mathbf{x}(f, n), \quad (4)$$

where the Wiener filter  $\mathbf{W}_j(f, n)$  is given by

$$\mathbf{W}_j(f, n) = v_j(f, n)\mathbf{R}_j(f) \left( \sum_{j'=1}^J v_{j'}(f, n)\mathbf{R}_{j'}(f) \right)^{-1}. \quad (5)$$

Finally, the time-domain source estimates  $\widehat{\mathbf{c}}_j(t)$  are recovered from  $\widehat{\mathbf{c}}_j(f, n)$  by inverse STFT.

Following this formulation, source separation becomes the problem of estimating the PSD and the spatial covariance matrices of each source. This can be achieved using an EM algorithm.

### 2.3 General iterative EM framework

The general iterative EM algorithm is summarized in Algorithm 1. It can be divided into the E-step and the M-step. The estimated PSDs  $v_j(f, n)$  are initialized in the *spectrogram initialization* step, for instance by computing the PSD of the mixture, while the estimated spatial covariance matrices  $\mathbf{R}_j(f)$  can be initialized by  $I \times I$  identity matrices. In the E-step, given the estimated parameters  $v_j(f, n)$  and  $\mathbf{R}_j(f)$  of each source, the source image estimates  $\widehat{\mathbf{c}}_j(f, n)$  are obtained by multichannel Wiener filtering (4) and the posterior second-order raw moments of the spatial source images  $\widehat{\mathbf{R}}_{\mathbf{c}_j}(f, n)$  are computed as

$$\widehat{\mathbf{R}}_{\mathbf{c}_j}(f, n) = \widehat{\mathbf{c}}_j(f, n)\widehat{\mathbf{c}}_j^H(f, n) + (\mathbf{I} - \mathbf{W}_j(f, n))v_j(f, n)\mathbf{R}_j(f), \quad (6)$$

where  $\mathbf{I}$  denotes the  $I \times I$  identity matrix and  $\cdot^H$  is the Hermitian transposition. In the M-step, the spatial covariance matrices  $\mathbf{R}_j(f)$  are updated as

$$\mathbf{R}_j(f) = \frac{1}{N} \sum_{n=1}^N \frac{1}{v_j(f, n)} \widehat{\mathbf{R}}_{\mathbf{c}_j}(f, n). \quad (7)$$

The source PSDs  $v_j(f, n)$  are first estimated without constraints as

$$z_j(f, n) = \frac{1}{I} \text{tr} \left( \mathbf{R}_j^{-1}(f) \widehat{\mathbf{R}}_{\mathbf{c}_j}(f, n) \right), \quad (8)$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix. Then, they are updated according to a given spectral model by fitting  $v_j(f, n)$  from  $z_j(f, n)$  in the *spectrogram fitting* step. The spectrogram initialization and the spectrogram fitting steps depend on how the spectral parameters are modeled. Spectral models used in this context may include NMF [24], which is a linear model with nonnegativity constraints, KAM [27], which relies on the local regularity of the sources, and continuity models [31]. In this study, we propose to use DNNs for this purpose.

## 3 DNN-based multichannel source separation

In this section, we propose a DNN-based multichannel source separation algorithm, which is based on the iterative algorithm presented in Section 2. Theoretical arguments regarding the cost function for DNN training are also presented.

### 3.1 Algorithm

In our algorithm, DNNs are employed to model the source spectra  $v_j(f, n)$ . We use them to predict the source spectra instead of the time-frequency masks because our preliminary experiments showed that the performance of both approaches was similar on our dataset. Moreover, it is more convenient to integrate DNNs that estimate spectra into Algorithm 1 because the algorithm requires PSD and the power spectrum can be viewed as an estimate of the PSD [32].

A DNN is used for spectrogram initialization and one or more DNNs are used for spectrogram fitting. Let  $\text{DNN}_0$  be the DNN used for spectrogram initialization and  $\text{DNN}_l$  the ones used for



**Algorithm 1** General iterative EM algorithm [23, 24]**Input:**

$\mathbf{x}(f, n)$  ▷ STFT of mixture  
 $J$  ▷ No. of sources  
 $L$  ▷ No. of EM iterations

```

1: for  $j \leftarrow 1, J$  do
2:    $v_j(f, n) \leftarrow$  spectrogram initialization
3:    $\mathbf{R}_j(f, n) \leftarrow I \times I$  identity matrix
4: end for
5: for  $l \leftarrow 1, L$  do
6:   for  $j \leftarrow 1, J$  do
7:      $\hat{\mathbf{c}}_j(f, n) \leftarrow$  Eq. (4)
8:      $\tilde{\mathbf{R}}_{\mathbf{c}_j}(f, n) \leftarrow$  Eq. (6)
9:      $\mathbf{R}_j(f) \leftarrow$  Eq. (7)
10:     $z_j(f, n) \leftarrow$  Eq. (8)
11:     $v_j(f, n) \leftarrow$  spectrogram fitting of  $z_j(f, n)$ 
12:   end for
13: end for
14: for  $j \leftarrow 1, J$  do
15:    $\hat{\mathbf{c}}_j(f, n) \leftarrow$  Eq. (4)
16: end for

```

**Output:**

$\hat{\mathbf{c}}_j(f, n)$  ▷ STFT of sources

spectrogram fitting.  $\text{DNN}_0$  aims to estimate the source spectra simultaneously from the observed mixture. This usage of joint DNN is similar to the usage of DNNs in the context of single-channel source separation in [12, 14, 15]. Meanwhile,  $\text{DNN}_l$  aims to improve the source spectra estimated at iteration  $l$ . This usage of DNN to obtain clean spectra from noisy spectra is similar to the usage of DNNs in the context of single-channel speech enhancement in [33, 34]. Theoretically, we can train different DNNs for spectrogram fitting at different iterations. Thus, the maximum number of DNNs for spectrogram fitting is equal to the number of iterations  $L$ .

In this research report, we consider magnitude STFT spectra as the input and output of DNNs. Following [19], the input and output spectra are computed from single-channel signals  $\tilde{x}(f, n)$  and  $\tilde{c}_j(f, n)$  obtained from the corresponding multichannel signals  $\mathbf{x}(f, n)$  and  $\mathbf{c}_j(f, n)$ , respectively, by DS beamforming. All DNNs are trained with the magnitude spectra of the single-channel source images  $|\tilde{c}_j(f, n)|$  as the target.

The inputs of  $\text{DNN}_0$  and  $\text{DNN}_l$  are denoted by  $|\tilde{x}(f, n)|$  and  $\sqrt{z_j(f, n)}$ , respectively. The outputs of both types of DNNs for source  $j$ , frequency bin  $f$ , and frame index  $n$  are denoted by  $\sqrt{v_j(f, n)}$ .  $\text{DNN}_0$  takes the magnitude spectra  $|\tilde{x}(f, n)|$  and yields the initial magnitude spectra  $\sqrt{v_j(f, n)}$  for all sources simultaneously.  $\text{DNN}_l$  takes the magnitude spectra  $\sqrt{z_j(f, n)}$  of all sources and yields the improved magnitude spectra  $\sqrt{v_j(f, n)}$  for all sources simultaneously.

The proposed DNN-based iterative algorithm is described in Algorithm 2.

**Algorithm 2** DNN-based iterative algorithm**Input:**

$\mathbf{x}(f, n)$	▷ STFT of mixture
$J$	▷ No. of sources
$K$	▷ No. of spatial updates
$L$	▷ No. of EM iterations
$\text{DNN}_0, \text{DNN}_1, \dots, \text{DNN}_L$	▷ DNN spectral models

```

1:  $\tilde{\mathbf{x}}(f, n) \leftarrow$  DS beamforming of  $\mathbf{x}(f, n)$ 
2:  $[v_1(f, n), v_2(f, n), \dots, v_J(f, n)] \leftarrow \text{DNN}_0 (|\tilde{\mathbf{x}}(f, n)|)^2$ 
3: for  $j \leftarrow 1, J$  do
4:    $\mathbf{R}_j(f, n) \leftarrow I \times I$  identity matrix
5: end for
6: for  $l \leftarrow 1, L$  do
7:   for  $k \leftarrow 1, K$  do
8:     for  $j \leftarrow 1, J$  do
9:        $\hat{\mathbf{c}}_j(f, n) \leftarrow$  Eq. (4)
10:       $\widehat{\mathbf{R}}_{\mathbf{c}_j}(f, n) \leftarrow$  Eq. (6)
11:       $\mathbf{R}_j(f) \leftarrow$  Eq. (7)
12:     end for
13:   end for
14:   for  $j \leftarrow 1, J$  do
15:      $z_j(f, n) \leftarrow$  Eq. (8)
16:   end for
17:    $[v_1(f, n), v_2(f, n), \dots, v_J(f, n)] \leftarrow$ 
      $\text{DNN}_l \left( \left[ \sqrt{z_1(f, n)}, \sqrt{z_2(f, n)}, \dots, \sqrt{z_J(f, n)} \right] \right)^2$ 
18: end for
19: for  $j \leftarrow 1, J$  do
20:    $\widehat{\mathbf{c}}_j(f, n) \leftarrow$  Eq. (4)
21: end for

```

**Output:**

$\widehat{\mathbf{c}}_j(f, n)$	▷ STFT of sources
--------------------------------	-------------------

### 3.2 Cost functions

We are interested in the use of different cost functions for training the DNNs.

1. The *Itakura-Saito (IS) divergence* [35] between the target  $|\tilde{c}_j(f, n)|$  and the estimate  $\sqrt{v_j(f, n)}$  is expressed as

$$\mathcal{D}_{\text{IS}} = \frac{1}{JFN} \sum_{j, f, n} \left( \frac{|\tilde{c}_j(f, n)|^2}{v_j(f, n)} - \log \frac{|\tilde{c}_j(f, n)|^2}{v_j(f, n)} - 1 \right). \quad (9)$$

It is a popular metric in the speech processing community because it yields signals with good perceptual quality. Moreover, it is desirable from a theoretical point of view because it results in maximum likelihood (ML) estimation of the spectra [35] and the whole Algorithm 2 then achieves ML estimation. While the IS divergence has become a popular

choice for NMF-based audio source separation [35–37], its use as the cost function for DNN training is uncommon.

2. The *Cauchy cost function* is expressed as

$$\mathcal{D}_{\text{Cau}} = \frac{1}{JFN} \sum_{j,f,n} \left( \frac{3}{2} \log (|\tilde{c}_j(f, n)|^2 + v_j(f, n)) - \log \sqrt{v_j(f, n)} \right). \quad (10)$$

It has been proposed recently for NMF-based audio source separation and advocated as performing better than the IS divergence in some cases [38].

3. The *phase-sensitive (PS) cost function* is defined as

$$\mathcal{D}_{\text{PS}} = \frac{1}{2JFN} \sum_{j,f,n} |m_j(f, n)\tilde{x}(f, n) - \tilde{c}_j(f, n)|^2, \quad (11)$$

where  $m_j(f, n) = v_j(f, n) / \sum_{j'} v_{j'}(f, n)$  is the single-channel Wiener filter [8, 22]. It minimizes the error in the complex-valued STFT domain, not in the magnitude STFT domain as the other cost functions considered here.

4. The *mean squared error (MSE)* [35] is expressed as

$$\mathcal{D}_{\text{MSE}} = \frac{1}{2JFN} \sum_{j,f,n} \left( |\tilde{c}_j(f, n)| - \sqrt{v_j(f, n)} \right)^2. \quad (12)$$

It is the most widely used cost function for various optimization processes, including DNN training for regression tasks. Despite its simplicity, it works well in most cases.

## 4 Experimental evaluation for speech enhancement

In this section, we present the application of the proposed framework for speech enhancement in the context of the CHiME-3 Challenge [39] and evaluate different design choices.

### 4.1 Task and dataset

The CHiME-3 Challenge is a speech separation and recognition challenge which considers the use of ASR for a multi-microphone tablet device. In this context, we consider two sources ( $J = 2$ ), namely speech and noise. The challenge provides real and simulated 6-channel microphone array data in 4 varied noise settings (bus, cafe, pedestrian area, and street junction) divided into training, development, and test sets. The training set consists of 1,600 real and 7,138 simulated utterances (`tr05_real` and `tr05_simu`), the development set consists of 1,640 real and 1,640 simulated utterances (`dt05_real` and `dt05_simu`), while the test set consists of 1,320 real and 1,320 simulated utterances (`et05_real` and `et05_simu`). The utterances are taken from the 5k vocabulary subset of the Wall Street Journal corpus [40]. All data are sampled at 16 kHz. For further details, please refer to [39].

We used the source separation performance metrics defined in BSS Eval toolbox 3.0<sup>1</sup> [41] in most of the experiments presented in this section. The metrics include signal to distortion ratio (SDR), source image to spatial distortion ratio (ISR), signal to interference ratio (SIR),

<sup>1</sup> [http://bass-db.gforge.inria.fr/bss\\_eval/](http://bass-db.gforge.inria.fr/bss_eval/)

and signal to artifacts ratio (SAR). In addition, at the end of this section, we use the best speech separation system as the front-end, combine it with the best back-end in [29], and evaluate the ASR performance in terms of word error rate (WER).

The ground truth speech and noise signals, which are employed as training targets for DNN-based speech enhancement, were extracted using the baseline simulation tool provided by the challenge organizers [39]. The ground truth speech and noise signals for the real data are not perfect because they are extracted based on an estimation of the impulse responses (IRs) between the close-talking microphone and the microphones on the tablet device. Hence, the resulting source separation performance metrics for the real data are unreliable. Therefore, we evaluate the separation performance on the simulated data for studying the impact of the different design choices. By contrast, since the ground truth transcriptions for ASR are reliable, we evaluate the ASR performance on real data.

## 4.2 General system design

The proposed DNN-based speech separation framework is depicted in Fig. 1. A single-channel variant of this framework which boils down to the approach in [19] is also depicted for comparison.

The framework can be divided into three main successive steps, namely pre-processing, spectrogram initialization, and multichannel filtering. We describe each step in detail below and then provide further description of the DNNs in the following section.

### 4.2.1 Preprocessing

The STFT coefficients were extracted using a Hamming window of length 1024 and hopsize 512 resulting  $F = 513$  frequency bins.

The time-varying time difference of arrivals (TDOAs) between the speaker’s mouth and each of the microphones are first measured using the provided baseline speaker localization tool [39], which relies on a nonlinear variant of steered response power using the phase transform (SRP-PHAT) [42, 43]. All channels are then aligned with each other by shifting the phase of STFT of the input noisy signal  $\mathbf{x}(f, n)$  in all time-frequency bins  $(f, n)$  by the opposite of the measured delay. This preprocessing is required to satisfy the model in (2) which assumes that the sources do not move over time.

In addition, we obtain a single-channel signal by averaging the realigned channels together. The combination of time alignment and channel averaging is known as DS beamforming in the microphone array literature [44, 45].

### 4.2.2 Spectrogram initialization

The initial PSDs of speech and noise are computed from the magnitude source spectra estimated by  $\text{DNN}_0$ .

### 4.2.3 Multichannel filtering

The PSDs and spatial covariance matrices of speech and noise are estimated and updated using the iterative algorithm (Algorithm 2), in which  $\text{DNN}_l$  is employed for spectrogram fitting at iteration  $l$ . In order to avoid numerical instabilities due to the use of single precision, the PSDs  $v_j(f, n)$  are floored to  $10^{-5}$  in the EM iteration.

In addition, the channels of estimated speech spatial image are averaged to obtain a single-channel signal for the ASR evaluation. Empirically, this provided better ASR performance than the use of one of the channels.

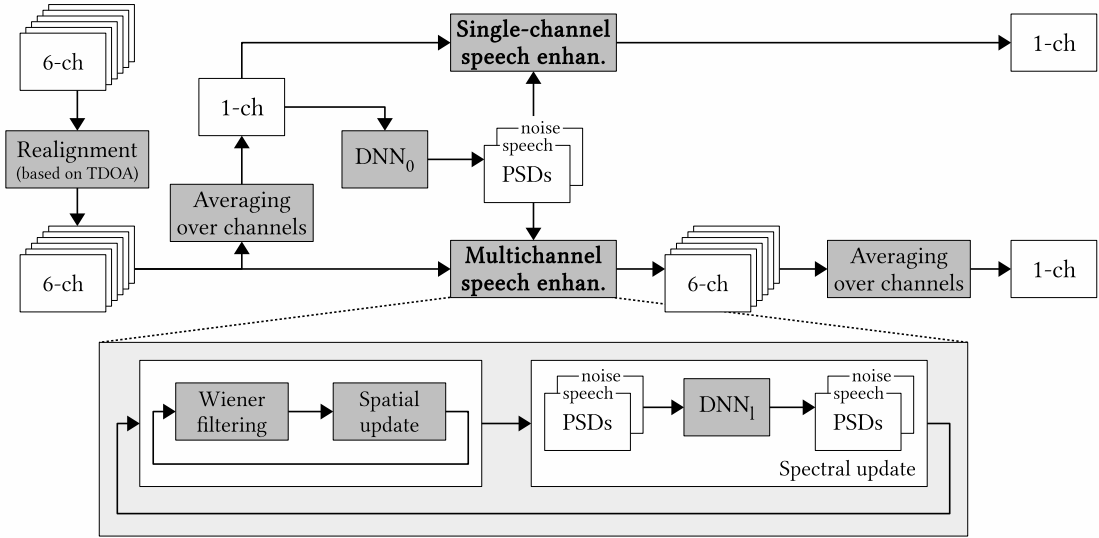


Figure 1: Proposed DNN-based speech separation framework. Both the single-channel and the multichannel versions are shown.

The number of spatial updates  $K$  is investigated in Section 4.5 and the number of iterations  $L$  in Section 4.6.

### 4.3 DNN spectral models

Three design aspects are discussed below: the architecture, the input and output, and the training.

#### 4.3.1 Architecture

The DNNs follow a multilayer perceptron (MLP) architecture. The number of hidden layers and the number of units in each input or hidden layer may vary. The number of units in the output layer equals the dimension of spectra multiplied by the number of sources. The activation functions of the hidden and output layers are rectified linear units (ReLUs) [46].

In this research report,  $DNN_0$  and  $DNN_l$  have a similar architecture. They have an input layer, three hidden layers, and an output layer. Both types of DNNs have hidden and output layers size of  $F \times J = 1026$ .  $DNN_0$  has an input layer sizes of  $F = 513$  and  $DNN_l$  of  $F \times J = 1026$ .

#### 4.3.2 Inputs and outputs

In order to provide temporal context, the input frames are concatenated into *supervectors* consisting of a center frame, left context frames, and right context frames. In choosing the context frames, we use every second frame relative to the center frame in order to reduce the redundancies caused by the windowing of STFT. Although this causes some information loss, this enables the supervectors to represent a longer context [16, 47]. In addition, we do not use the magnitude spectra of the context frames directly, but the difference of magnitude between the context frames spectra and the center frame. These differences act as complementary features similar to delta features.

Let  $|\tilde{x}(f, n)|$  be the input frames of  $\text{DNN}_0$ . The supervector can be expressed as

$$Z_0(f, n) = \begin{bmatrix} |\tilde{x}(f, n - 2c)| - |\tilde{x}(f, n)| \\ \vdots \\ |\tilde{x}(f, n)| \\ \vdots \\ |\tilde{x}(f, n + 2c)| - |\tilde{x}(f, n)| \end{bmatrix} \quad (13)$$

where  $c$  is the one-sided context length in frames. The supervector for  $\text{DNN}_l$ ,  $Z_l(f, n)$ , is constructed in a similar way where a stack of  $\sqrt{z_j(f, n)}$  is used as input instead of  $|\tilde{x}(f, n)|$  (see Fig. 2 and 3). In this research report, we considered  $c = 2$ , so the supervectors for the input of the DNNs were composed by 5 time frames (2 left context, 1 center, and 2 right context frames).

The dimension of the supervectors is reduced by principal component analysis (PCA) to the dimension of the DNN input. This dimensionality reduction is done to reduce the computational cost, especially when training the DNNs. Standardization (zero mean, unit variance) is done element-wise before and after PCA over the training data as in [48]. The standardization factors and the PCA transformation matrix are then kept for pre-processing of any input. Thus, strictly speaking, the inputs of DNNs are not the supervectors of magnitude spectra  $Z_0(f, n)$  and  $Z_l(f, n)$ , but their transformation into reduced dimension vectors.

Fig. 2 and 3 illustrates the inputs and outputs of the DNNs for spectrogram initialization and spectrogram fitting, respectively.  $F$  denotes the dimension of the spectra,  $C = 2c + 1$  the context length, and  $J$  the number of sources.

#### 4.3.3 Training criterion

The cost function used for DNN training is the sum of a primary cost function and an  $\ell_2$  regularization term. The  $\ell_2$  regularization term [49] is used to prevent overfitting and can be expressed as

$$\mathcal{D}_{\ell_2} = \frac{\lambda}{2} \sum_q w_q^2 \quad (14)$$

where  $w_q$  are the DNN weights and the regularization parameter is fixed to  $\lambda = 10^{-5}$ . No regularization is applied to the biases.

Table 1 summarizes the implementation of different cost functions for the experiments. In order to avoid numerical instabilities, instead of using the original formulation of IS divergence in (9), our implementation used a regularized formulation as shown in (15). It should be noted that the use of regularization in this case is a common practice to avoid instabilities [36, 50]. For the same reason, we used a regularized formulation of Cauchy cost function as shown in (16). For both divergences, the regularization parameter is set to  $\delta = 10^{-3}$ . In addition, geometric analysis on the PS cost function by considering that  $m_j(f, n) \in \mathbb{R}_+^{F \times N}$  leads to a simplified formula shown in (17).

#### 4.3.4 Training algorithm

The weights are initialized randomly from a zero-mean Gaussian distribution with standard deviation of  $\sqrt{2/n_l}$ , where  $n_l$  is the fan-in (the number of inputs to the neuron, which is equal to the size of the previous layer in our case) [51]. Finally, the biases are initialized to zero.

The DNNs are trained by greedy layer-wise supervised training [52] where the hidden layers are added incrementally. In the beginning, a NN with one hidden layer is trained after random

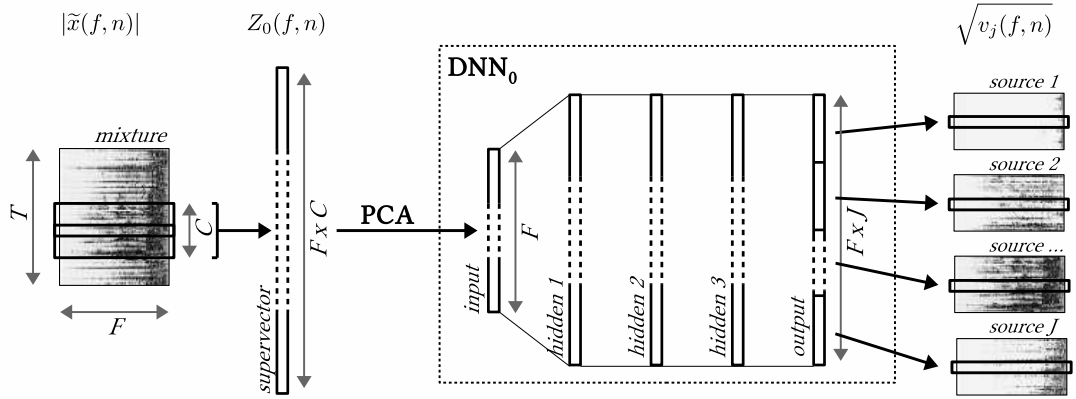


Figure 2: Illustration of the inputs and outputs of the DNN for spectrogram initialization. Input: magnitude spectrum of the mixture (left). Output: magnitude spectra of the sources (right).

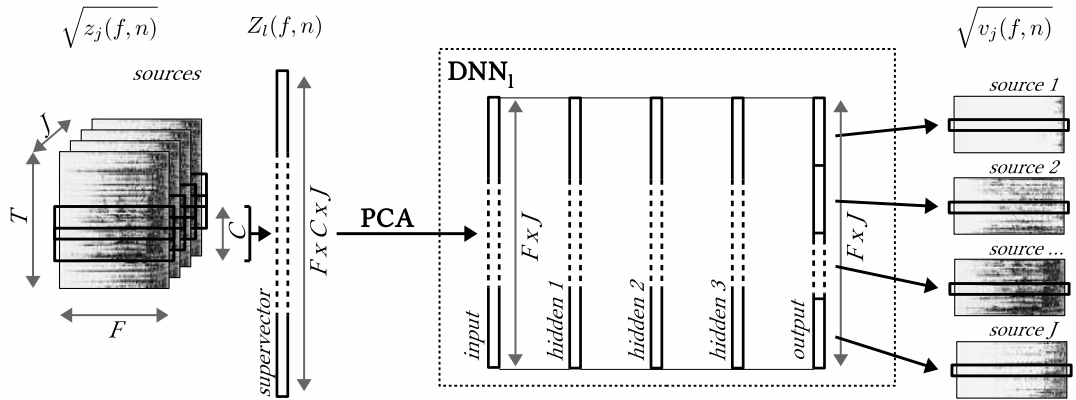


Figure 3: Illustration of the inputs and outputs of the DNNs for spectrogram fitting. Input: stack of magnitude spectra of all sources (left). Output: magnitude spectra of the sources (right).

weight initialization. The output layer of this trained NN is then substituted by new hidden and output layers to form a new NN, while the parameters of the existing hidden layer are kept. Thus, we can view this as a pre-training method for the training of a new NN. After random initialization for the parameters of new layers, the new NN is entirely trained. This procedure is done iteratively until the target number of hidden layers is reached.

Training is done by backpropagation with minibatch size of 100 and the ADADELTA parameter update algorithm [53]. Compared to standard stochastic gradient descent (SGD), ADADELTA employs adaptive per-dimension learning rates and does not require manual setting of the learning rate. The hyperparameters of ADADELTA are set to  $\rho = 0.95$  and  $\epsilon = 10^{-6}$  following [53]. The validation error is computed every epoch and the training is stopped after 10 consecutive epochs failed to obtain better validation error. The latest model which yields the best validation error is kept. Besides, the maximum number of training epochs is set to 100.

The DNNs for the source separation evaluation were trained on both the real and simulated

Table 1: Implementation details of the DNN training cost functions.

Exp. label	Weight reg.	Primary cost function
IS	$\mathcal{D}_{\ell_2}$	$\mathcal{D}_{\text{IS}} = \frac{1}{JFN} \sum_{j,f,n} \left( \frac{ \tilde{c}_j(f,n) ^2 + \delta}{v_j(f,n) + \delta} - \log( \tilde{c}_j(f,n) ^2 + \delta) + \log(v_j(f,n) + \delta) - 1 \right)$ (15)
Cau	$\mathcal{D}_{\ell_2}$	$\mathcal{D}_{\text{Cau}} = \frac{1}{JFN} \sum_{j,f,n} \left( \frac{3}{2} \log( \tilde{c}_j(f,n) ^2 + v_j(f,n) + \delta) - \log\left(\sqrt{v_j(f,n) + \delta}\right) \right)$ (16)
PS	$\mathcal{D}_{\ell_2}$	$\mathcal{D}_{\text{PS}} = \frac{1}{2JFN} \sum_{j,f,n} \left( \frac{v_j(f,n)}{\sum_{j'} v_{j'}(f,n)}  \tilde{x}(f,n)  -  \tilde{c}_j(f,n)  \cos(\angle \tilde{x}(f,n) - \angle \tilde{c}_j(f,n)) \right)^2$ (17)
MSE	$\mathcal{D}_{\ell_2}$	$\mathcal{D}_{\text{MSE}} = \frac{1}{2JFN} \sum_{j,f,n} \left(  \tilde{c}_j(f,n)  - \sqrt{v_j(f,n)} \right)^2$ (12)

training sets (`tr05_real` and `tr05_simu`) with the real and simulated development sets (`dt05_real` and `dt05_simu`) as validation data. Conversely, we trained the DNNs for the speech recognition evaluation on the real training set only (`tr05_real`) and validated them on the real development set only (`dt05_real`). See [29] for the performance comparison between these two different settings.

#### 4.4 Impact of cost functions

We first evaluated the impact of the cost function by setting  $L = 0$  (see Algorithm 2) so that the separation relied on the PSD estimates  $v_j(f, n)$  by letting the spatial covariance matrices  $\mathbf{R}_j(f)$  be the identity matrix. This is equivalent to single-channel source separation for each channel.

Fig. 4 shows the evaluation results for the resulting 6-channel estimated speech signal on the simulated test set (`et05_simu`).

According to most metrics, 'MSE' and 'PS' worked better than 'IS' and 'Cau'. The performance of 'MSE' and 'PS' was similar on average. In accordance with the results in [38], 'Cau' was better than 'IS' in most cases. 'IS' and 'Cau' have poor SIR properties. Interestingly, 'Cau' outperformed the others in term of SAR.

In addition, we mention here that the use of flooring function (e.g. ReLU activation function for the DNN outputs) during the training with 'IS', 'Cau', 'PS' seems to be important. We found in additional experiments (not shown here) that training failed when a linear activation function was used for the output layer with these three cost functions.

#### 4.5 Impact of spatial parameters updates

In this subsection, we investigate the impact of the spatial parameters updates on the performance by setting the number of iterations to  $L = 1$  and varying the number of spatial updates  $K$ , while ignoring the computation of  $z_j(f, n)$  and the spectral parameters update (lines 14–17 of Algorithm 2). Thus, the spectral parameters  $v_j(f, n)$  are only initialized by the first DNN (as in Section 3.2) and kept fixed during the iterative procedure. We evaluate the different cost functions from Section 3.2 in this context again.



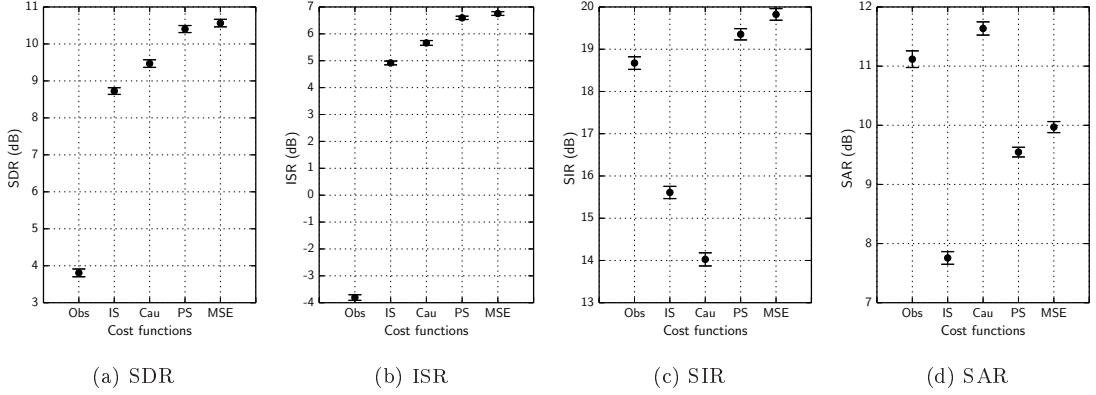


Figure 4: Performance comparison for the DNNs trained with different cost functions. The PSDs  $v_j(f, n)$  are estimated by  $\text{DNN}_0$  and the spatial covariance matrices  $\mathbf{R}_j(f)$  are the identity matrix. 'Obs' represents the performance measured on the observed 6-channel mixture signal. The evaluation was done on the simulated test set (`et05_simu`). The figures show the mean value and the 95% confidence interval. Higher is better.

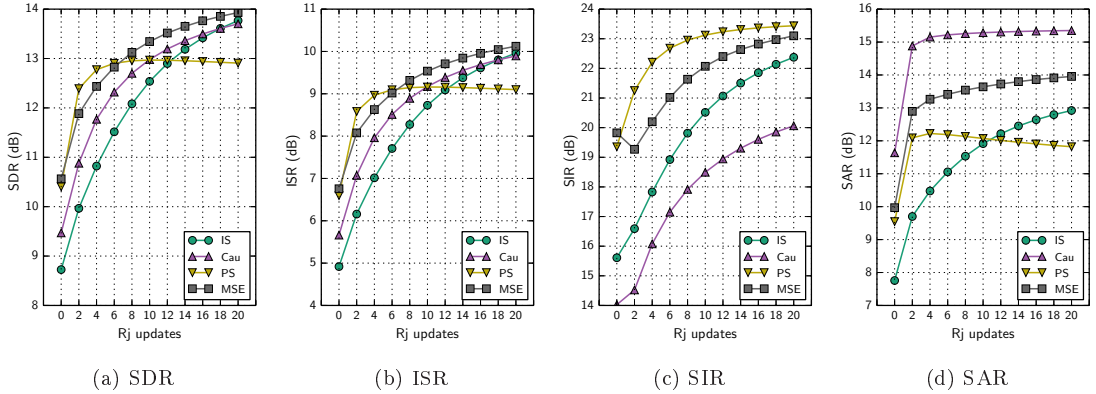


Figure 5: Performance comparison for various numbers of spatial updates with the DNNs trained with different cost functions. The PSDs  $v_j(f, n)$  are estimated by the  $\text{DNN}_0$  and the spatial covariance matrices  $\mathbf{R}_j(f)$  are updated in the iterative procedure. The evaluation was done on the simulated test set (`et05_simu`). The figures show the mean value. The 95% confidence intervals are similar to those in Fig. 4. Higher is better.

Fig. 5 shows the results for the resulting 6-channel estimated speech signal on the simulated test set (`et05_simu`). The x-axis of each chart corresponds to the number of spatial updates  $k$ . Thus,  $k = 0$  is equivalent to single-channel source separation for each channel whose results are shown in Fig. 4.

In general, the performance of 'PS' saturated after a few updates, while the performance of other cost functions increased with  $k$ . Interestingly, after 20 iterations, 'IS' and 'Cau' performed better than 'PS' and similarly to 'MSE' in terms of SDR and ISR. The SIR and SAR of most cost functions were low when  $\mathbf{R}_j(f)$  is set to the identity matrix (as also shown in Fig. 4), they

recovered and got better along the spatial updates, except the SAR of 'PS' which decreased after a few updates.

In summary, the proposed multichannel approach outperformed single-channel DNN-based approach even when using  $DNN_0$  only. The spatial parameters and their updates improved the enhancement performance. From the experiments using 20 spatial parameter updates, the SDR and ISR improved significantly for most cost functions. 'MSE' is the most reasonable choice because it improved all of the metrics well. 'PS' is suitable for the tasks that put emphasis on the SIR instead of the SAR. On the contrary, 'Cau' is suitable for the tasks that put emphasis on the SAR instead of the SIR. Finally, 'IS' should be avoided.

#### 4.6 Impact of spectral parameters updates

In this subsection, we investigate the impact of spectral parameter updates (i.e. the spectrogram fitting) on the performance by setting the number of spatial updates to  $K = 20$ , varying the number of iterations  $L$ , and varying the DNN used for iteration  $l$ . We also evaluate different cost functions in this context, namely IS, Cauchy, and MSE. We left the PS cost function because as shown previously, its SDR performance after 20 spatial updates was significantly lower than the others.

We trained two additional DNNs ( $DNN_1$  and  $DNN_2$ ) for spectrogram fitting. This allowed us to try different settings for the iterative procedure: (1) without spectral updates; (2) with spectral updates using only  $DNN_1$ ; and (3) with spectral updates using  $DNN_1$  and  $DNN_2$ .

We present the comparison of these three settings using MSE as the cost function in Fig. 6. We then present the comparison of different cost functions using the third setting in Fig. 7. For both figures, the x-axis shows the index of EM iteration  $l$ , the update type (spatial or spectral), and the DNN index. Thus,  $l = 0$  is equivalent to single-channel source separation for each channel whose results are shown in Fig. 4, while  $l = 1$  with spatial updates is equivalent to the results shown in Fig. 5.

Fig. 6 shows that the use of a specific DNN for each iteration (here,  $DNN_1$  for  $l = 1$  and  $DNN_2$  for  $l = 2$ ) is important. When a specific DNN is used, the spectral update provides a small improvement. Most importantly, this update allows the following spatial update to yield significant improvement. This behavior can be observed by comparing the performance of the spectral updates of EM iteration  $l$  and the spatial updates of the following iteration  $l + 1$ . Additionally, we can observe it by comparing the overall behavior of the "3 DNNs" curve to the "1 DNN" curve, in which no spectrogram fitting is done. Fig. 7 shows similar behavior for the other cost functions.

Fig. 6 also shows that the use of the same DNN for several iterations (here,  $DNN_1$  for  $l = 1$  and  $l = 2$ ) has detrimental effect on the performance. We can observe this by comparing the "3 DNNs" curve to the "2 DNNs" curve for  $l = 2$  and  $l = 3$ . It is understandable because there is a mismatch between the input and the training data of the DNN in this case.

In summary, the iterative spectral and spatial updates improve the enhancement performance. The performance saturates after few EM iteration. MSE performs better than the other cost functions. Although the use of IS divergence or Cauchy cost function for DNN training is theoretically motivated, the resulting performance is lower than with MSE.

#### 4.7 Speech recognition

In this subsection, we evaluate the use of our best system as the front-end of a speech recognition system. We did a speech recognition evaluation by following the Kaldi setup distributed by the

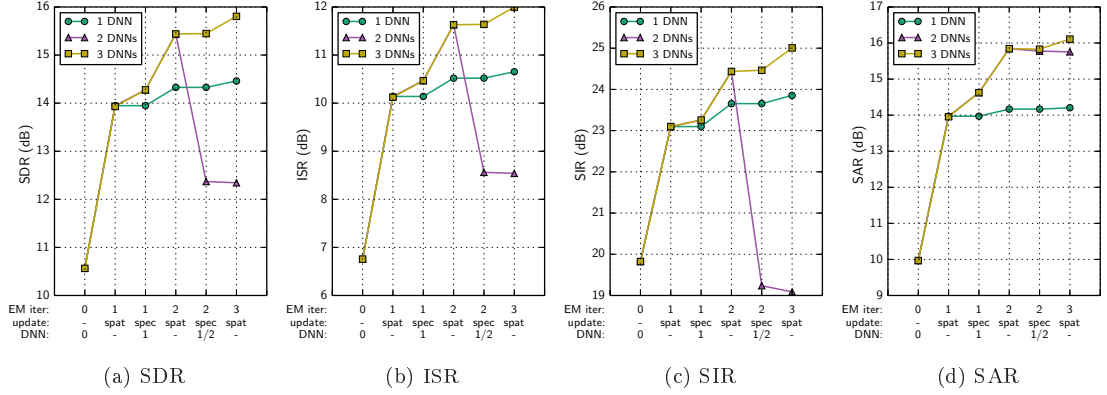


Figure 6: Performance comparison for each update of the EM iterations in which different number of DNNs are used. In "1 DNN", there is no spectrogram fitting. In "2 DNNs", DNN<sub>1</sub> is used for spectrogram fitting of both  $l = 1$  and  $l = 2$ . In "3 DNNs", DNN<sub>1</sub> and DNN<sub>2</sub> are used for spectrogram fitting of  $l = 1$  and  $l = 2$ , respectively. Some markers and lines are not visible because they coincide. The DNNs are trained with MSE. The spatial covariance matrices  $\mathbf{R}_j(f)$  are updated with  $K = 20$ . The evaluation was done on the simulated test set (et05\_simu). The figures show the mean value. The 95% confidence intervals are similar to those in Fig. 4. Higher is better.

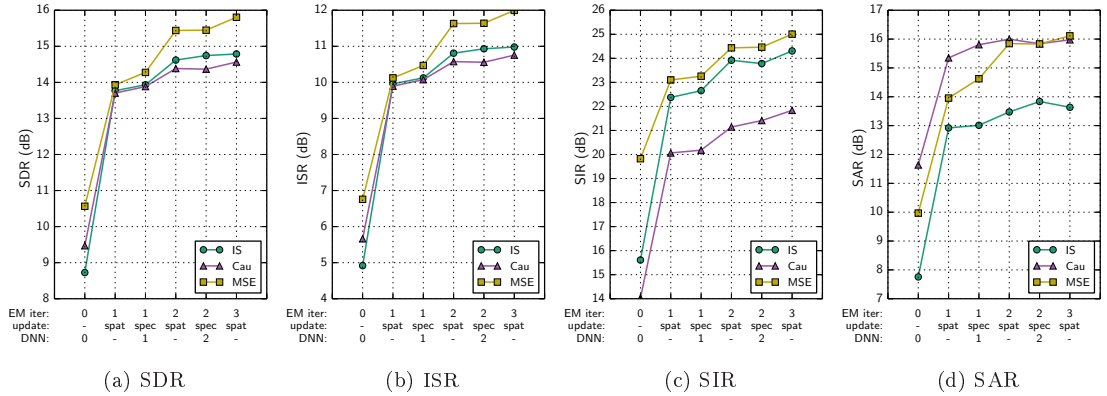


Figure 7: Performance comparison for each update of the EM iterations with the DNNs trained with different cost functions. Different DNNs are used for each EM iteration. The spatial covariance matrices  $\mathbf{R}_j(f)$  are updated with  $K = 20$ . The evaluation was done on the simulated test set (et05\_simu). The figures show the mean value. The 95% confidence interval for each cost function is similar to the interval of corresponding cost function in Fig. 4. Higher is better.

CHiME-3 challenge organizers<sup>2</sup> [39, 54]. The evaluation includes the uses of (a) feature-space maximum likelihood regression (fMLLR) features [55]; (b) acoustic models based on Gaussian Mixture Model (GMM) and DNN trained with the cross entropy (CE) criterion followed by state-level minimum Bayes risk (sMBR) criterion [56]; and (c) language models with 5-gram

<sup>2</sup><https://github.com/kaldi-asr/kaldi/tree/master/egs/chime3>

Table 2: Average WERs (%) using the GMM back-end retrained on enhanced multi-condition data. The evaluation was done on the real sets. Lower is better.

Enhancement method	EM iter.	Update type	Dev	Test
Observed	-	-	18.32	33.02
DS beamforming	-	-	14.07	25.86
Proposed: MSE (3 DNNs)	0	-	13.94	25.97
	1	spatial	<b>11.97</b>	<b>22.24</b>
		spectral	12.73	23.84
	2	spatial	12.43	24.28
		spectral	12.02	24.27
	3	spatial	12.55	24.83

Table 3: Average WERs (%) using the DNN+sMBR back-end trained with enhanced multi-condition data followed by 5-gram KN smoothing and RNN-LM rescoring. The evaluation was done on the real sets. Lower is better.

Enhancement method	EM iter.	Update type	Dev	Test
Observed	-	-	9.65	19.28
DS beamforming	-	-	6.35	13.70
Proposed: MSE (3 DNNs)	0	-	6.91	14.87
	1	spatial	<b>5.59</b>	<b>12.77</b>
		spectral	6.36	13.66
	2	spatial	5.92	14.14
		spectral	5.77	13.73
	3	spatial	6.07	14.95

Kneser-Ney (KN) smoothing [57] and rescoring using recurrent neural network-based language model (RNN-LM) [58]. The acoustic models are trained on enhanced multi-condition real and simulated data. The evaluation results are presented in terms of word error rate (WER). The optimization of the speech recognition back-end is beyond the scope of this research report. Please refer to [54] for the further details of the methods used in the evaluation.

The evaluation results include the baseline performance (observed) and DS beamforming. The baseline performance was measured using only channel 5 of the observed signal. This channel is considered as the most useful channel because the corresponding microphone faces the user and is located at the bottom-center of the tablet device. DS beamforming was performed on the 6-channel observed signal as described in Section 4.2.

Table 2 shows the performance comparison using the GMM back-end retrained on enhanced multi-condition data. Table 3 shows the performance comparison using the DNN+sMBR back-end trained with enhanced multi-condition data followed by 5-gram KN smoothing and RNN-LM rescoring. Boldface numbers show the best performance for each dataset.

For the single-channel enhancement (EM iteration  $l = 0$ ), the WER on the real test set decreases by 12% and 19% relative using the GMM and the DNN+sMBR backends, respectively. Interestingly, this single-channel enhancement which is done after DS beamforming did

not provide better performance compared to the DS beamforming alone. It indicates that proper exploitation of multichannel information is crucial.

The proposed multichannel enhancement using the DNNs trained on real data only then decreases the WER on the real test set by 14% relative using both the GMM and the DNN+sMBR backends w.r.t. the corresponding single-channel enhancement. It also decreases the WER on the real test set by 14% and 7% relative using the GMM and the DNN+sMBR backends, respectively, w.r.t. the DS beamforming alone.

## 5 Conclusion

In this research report, we presented a DNN-based multichannel source separation framework where the multichannel filter is derived from the source spectra, which are estimated by DNNs, and the spatial covariance matrices, which are updated iteratively in an EM fashion. Evaluation has been done for a speech enhancement task. The experimental results show that the proposed framework works well and outperforms single-channel DNN-based enhancement. The use of a single DNN to estimate the source spectra from the mixture already suffices to observe an improvement. Spectral updates by employing additional DNNs moderately improve the performance themselves, but they allow the following spatial updates to provide further significant improvement. We also demonstrate that the use of a specific DNN for each iteration is beneficial. Finally, the use of MSE as the DNN training cost function is shown to provide the best performance.

Future directions concern alternative training targets for DNNs, the use of spatial features [9–11] as additional inputs, and the incorporation of prior information about the source position.

## Acknowledgment

The authors would like to thank the developers of Theano [59, 60] and Kaldi [61]. Experiments presented in this research report were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

## References

- [1] S. Makino, H. Sawada, and T.-W. Lee, Eds., *Blind Speech Separation*, ser. Signals and Communication Technology. Dordrecht, The Netherlands: Springer, 2007.
- [2] M. Wölfel and J. McDonough, *Distant Speech Recognition*. Chichester, West Sussex, UK: Wiley, 2009.
- [3] T. Virtanen, R. Singh, and B. Raj, Eds., *Techniques for Noise Robustness in Automatic Speech Recognition*. Chichester, West Sussex, UK: Wiley, 2012.
- [4] G. R. Naik and W. Wang, Eds., *Blind Source Separation: Advances in Theory, Algorithms and Applications*, ser. Signals and Communication Technology. Berlin, Germany: Springer, 2014.
- [5] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, “From blind to guided audio source separation: How models and side information can improve the separation of sound,” *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 107–115, May 2014.

- 
- [6] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, ser. Found. Trends Signal Process. Hanover, MA, USA: Now Publishers Inc., Jun. 2014, vol. 7, no. 3-4.
- [7] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [8] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. Int'l. Conf. Latent Variable Analysis and Signal Separation*, Liberec, Czech Republic, Aug. 2015.
- [9] J. Chen, Y. Wang, and D. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1993–2002, Dec. 2014.
- [10] Y. Jiang, D. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 2112–2121, Dec. 2014.
- [11] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 116–120.
- [12] Y. Tu, J. Du, Y. Xu, L. Dai, and C.-H. Lee, "Speech separation based on improved deep neural networks with dual outputs of speech features for both target and interfering speakers," in *Proc. Int'l. Symp. Chinese Spoken Lang. Process. (ISCSLP)*, Singapore, Sept 2014, pp. 250–254.
- [13] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proc. Int'l. Soc. for Music Inf. Retrieval (ISMIR)*, Taipei, Taiwan, Oct. 2014, pp. 477–482.
- [14] —, "Deep learning for monaural speech separation," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 1562–1566.
- [15] —, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.
- [16] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 2135–2139.
- [17] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.
- [18] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Vancouver, Canada, May 2013, pp. 7092–7096.

- 
- [19] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, “Discriminatively trained recurrent neural networks for single-channel speech separation,” in *Proc. IEEE Global Conf. Signal and Information Process. (GlobalSIP)*, Atlanta, GA, USA, Dec. 2014, pp. 577–581.
- [20] A. Narayanan and D. Wang, “Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.
- [21] Y. Wang and D. Wang, “A deep neural network for time-domain signal reconstruction,” in *Proc. IEEE Int’l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 4390–4394.
- [22] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *Proc. IEEE Int’l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 708–712.
- [23] N. Q. K. Duong, E. Vincent, and R. Gribonval, “Under-determined reverberant audio source separation using a full-rank spatial covariance model,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1830–1840, Jul. 2010.
- [24] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1118–1133, May 2012.
- [25] T. Gerber, M. Dutasta, L. Girin, and C. Févotte, “Professionally-produced music separation guided by covers,” in *Proc. Int’l. Soc. for Music Inf. Retrieval (ISMIR)*, Porto, Portugal, Oct. 2012, pp. 85–90.
- [26] M. Togami and Y. Kawaguchi, “Simultaneous optimization of acoustic echo reduction, speech dereverberation, and noise reduction against mutual interference,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 11, pp. 1612–1623, Nov. 2014.
- [27] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, “Kernel additive models for source separation,” *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4298–4310, Aug. 2014.
- [28] A. Liutkus, D. Fitzgerald, and Z. Rafii, “Scalable audio separation with light kernel additive modelling,” in *Proc. IEEE Int’l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 76–80.
- [29] S. Sivasankaran, A. A. Nugraha, E. Vincent, J. A. Morales-Cordovilla, S. Dalmia, I. Illina, and A. Liutkus, “Robust ASR using neural network based speech enhancement and feature simulation,” in *Proc. IEEE Automat. Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, AZ, USA, Dec. 2015, pp. 482–489.
- [30] E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, “Probabilistic modeling paradigms for audio source separation,” in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. Hershey, PA, USA: IGI Global, 2011, ch. 7, pp. 162–185.
- [31] N. Duong, H. Tachibana, E. Vincent, N. Ono, R. Gribonval, and S. Sagayama, “Multichannel harmonic and percussive component separation by joint modeling of spatial and spectral continuity,” in *Proc. IEEE Int’l Conf. Acoust. Speech Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 205–208.

- [32] A. Liutkus and R. Badeau, "Generalized wiener filtering with fractional power spectrograms," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 266–270.
- [33] D. Liu, P. Smaragdis, and M. Kim, "Experiments on deep learning for speech denoising," in *Proc. ISCA INTERSPEECH*, Singapore, Sep. 2014, pp. 2685–2688.
- [34] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.
- [35] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830, Mar. 2009.
- [36] A. Lefèvre, F. Bach, and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence," in *Proc. IEEE Workshop on Appl. of Signal Process. to Audio and Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2011, pp. 313–316.
- [37] N. Bertin, C. Févotte, and R. Badeau, "A tempering approach for Itakura-Saito non-negative matrix factorization. with application to music transcription," in *Proc. IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 1545–1548.
- [38] A. Liutkus, D. Fitzgerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *Proc. IEEE Workshop on Appl. of Signal Process. to Audio and Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2015.
- [39] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Automat. Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, AZ, USA, Dec. 2015, pp. 504–511.
- [40] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Complete," 2007, Linguistic Data Consortium, Philadelphia.
- [41] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.
- [42] B. Loesch and B. Yang, "Adaptive segmentation and separation of determined convolutive mixtures under dynamic conditions," in *Proc. Int'l. Conf. Latent Variable Analysis and Signal Separation*, Saint-Malo, France, 2010, pp. 41–48.
- [43] C. Blandin, A. Ozerov, and E. Vincent, "Multi-source TDOA estimation in reverberant audio using angular spectra and clustering," *Signal Processing*, vol. 92, no. 8, pp. 1950–1960, 2012.
- [44] J. McDonough and K. Kumatani, "Microphone arrays," in *Techniques for Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. Chichester, West Sussex, UK: Wiley, 2012, ch. 6.
- [45] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 127–140, 2012.



- [46] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier networks,” in *Proc. Int’l. Conf. Artificial Intelligence and Statistics (AISTATS)*, vol. 15, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.
- [47] A. A. Nugraha, K. Yamamoto, and S. Nakagawa, “Single-channel dereverberation by feature mapping using cascade neural networks for robust distant speaker identification and speech recognition,” *EURASIP J. Audio, Speech and Music Process.*, vol. 2014, no. 13, 2014.
- [48] X. Jaureguiberry, E. Vincent, and G. Richard, “Fusion methods for audio source separation,” Dec. 2014. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01120685>
- [49] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. Orr, and K.-R. Müller, Eds. Berlin, Germany: Springer, 2012, vol. 7700, ch. 19, pp. 437–478.
- [50] P. Sprechmann, A. M. Bronstein, and G. Sapiro, “Supervised non-negative matrix factorization for audio source separation,” in *Excursions in Harmonic Analysis, Volume 4*, ser. Applied and Numerical Harmonic Analysis, R. Balan, M. Begué, J. J. Benedetto, W. Czaja, and K. A. Okoudjou, Eds. Switzerland: Springer, 2015, pp. 407–420.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *ArXiv e-prints*, Feb. 2015. [Online]. Available: <http://arxiv.org/abs/1502.01852>
- [52] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Proc. Conf. on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, Dec. 2006, pp. 153–160.
- [53] M. D. Zeiler, “ADADELTA: An adaptive learning rate method,” *ArXiv e-prints*, Dec. 2012. [Online]. Available: <http://arxiv.org/abs/1212.5701>
- [54] T. Hori, Z. Chen, H. Erdogan, J. R. Hershey, J. Le Roux, V. Mitra, and S. Watanabe, “The MERL/SRI system for the 3rd chime challenge using beamforming, robust feature extraction, and advanced speech recognition,” in *Proc. IEEE Automat. Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, AZ, USA, Dec. 2015, pp. 475–481.
- [55] M. J. F. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer Speech & Language*, vol. 12, no. 2, pp. 75 – 98, 1998.
- [56] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *Proc. ISCA INTERSPEECH*, Lyon, France, Aug. 2013, pp. 2345–2349.
- [57] R. Kneser and H. Ney, “Improved backing-off for M-gram language modeling,” in *Proc. IEEE Int’l Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, Detroit, MI, USA, May 1995, pp. 181–184.
- [58] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. ISCA INTERSPEECH*, Chiba, Japan, Sep. 2010, pp. 1045–1048.
- [59] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: a CPU and GPU math expression compiler,” in *Proc. of the Python for Scientific Computing Conference (SciPy)*, Austin, TX, USA, Jun. 2010.

- 
- [60] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, “Theano: new features and speed improvements,” Deep Learning and Unsupervised Feature Learning Neural Information Processing Systems (NIPS) Workshop, Lake Tahoe, CA, USA, 2012.
- [61] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi speech recognition toolkit,” in *Proc. IEEE Automat. Speech Recognition and Understanding Workshop (ASRU)*, Hawaii, USA, Dec. 2011.



## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Problem formulation . . . . .	4
2.2	Model . . . . .	4
2.3	General iterative EM framework . . . . .	5
<b>3</b>	<b>DNN-based multichannel source separation</b>	<b>5</b>
3.1	Algorithm . . . . .	5
3.2	Cost functions . . . . .	7
<b>4</b>	<b>Experimental evaluation for speech enhancement</b>	<b>8</b>
4.1	Task and dataset . . . . .	8
4.2	General system design . . . . .	9
4.2.1	Preprocessing . . . . .	9
4.2.2	Spectrogram initialization . . . . .	9
4.2.3	Multichannel filtering . . . . .	9
4.3	DNN spectral models . . . . .	10
4.3.1	Architecture . . . . .	10
4.3.2	Inputs and outputs . . . . .	10
4.3.3	Training criterion . . . . .	11
4.3.4	Training algorithm . . . . .	11
4.4	Impact of cost functions . . . . .	13
4.5	Impact of spatial parameters updates . . . . .	13
4.6	Impact of spectral parameters updates . . . . .	15
4.7	Speech recognition . . . . .	15
<b>5</b>	<b>Conclusion</b>	<b>18</b>



**RESEARCH CENTRE  
NANCY – GRAND EST**

615 rue du Jardin Botanique  
CS20101  
54603 Villers-lès-Nancy Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399