

Sparse Analysis Model Based Dictionary Learning for Signal Declipping

Bin Li, Lucas Rencker, Jing Dong, Yuhui Luo, *Member, IEEE*,
Mark D. Plumbley, *Fellow, IEEE*, and Wenwu Wang, *Senior Member, IEEE*

Abstract—Clipping is a common type of distortion in which the amplitude of a signal is truncated if it exceeds a certain threshold. Sparse representation has underpinned several algorithms developed recently for reconstruction of the original signal from clipped observations. However, these declipping algorithms are often built on a synthesis model, where the signal is represented by a dictionary weighted by sparse coding coefficients. In contrast to these works, we propose a sparse analysis-model-based declipping (SAD) method, where the declipping model is formulated on an analysis (i.e. transform) dictionary, and additional constraints characterizing the clipping process. The analysis dictionary is updated using the Analysis SimCO algorithm, and the signal is recovered by using a least-squares based method or a projected gradient descent method, incorporating the observable signal set. Numerical experiments on speech and music are used to demonstrate improved performance in signal to distortion ratio (SDR) compared to recent state-of-the-art methods including A-SPADE and ConsDL.

Index Terms—ASimCO, sparse analysis, clipping signal, nonlinear measurement

I. INTRODUCTION

SPARSE representation plays an important role in inverse problems and signal recovery tasks, such as denoising, inpainting, declipping and super-resolution. By exploiting the sparsity of signals in some domain, such as a transform domain, the original signals can be estimated from the observed signals using a sparse representation model. Clipping is a common type of distortion whereby the amplitude of a signal is truncated if it exceeds a certain threshold. Clipping may occur when a signal is recorded by a sensor limited by the range of data that can be physically measured, when the signal is digitized, or when an analog or digital signal is transformed

through a nonlinear system. The clipped signal $y \in \mathbb{R}^m$ is a distorted observation of the ground truth signal $x \in \mathbb{R}^m$,

$$y = f(x) \quad (1)$$

where f is a nonlinear clipping function. The observed signal y may be divided into clipped regions and reliable (i.e. unclipped) regions. Declipping aims to recover the original signal x from the clipped signal y . This can be achieved with a variety of methods, as summarized in a recent survey paper [1]. Typically, this problem can be formulated using a cost function based on reconstruction errors, together with constraints or regularizers that are consistent with the clipping process [2], [3], [4], [5], [6].

A popular model used for the declipping problem is the sparse synthesis model [7], [8]. In this model, the original signal x is represented using an overcomplete dictionary $\Omega^T \in \mathbb{R}^{m \times p}$, i.e. $x = \Omega^T \alpha$, where $\alpha \in \mathbb{R}^p$ is a sparse coefficient vector, $p > m$, and the superscript T denotes matrix transpose. The original signal x can be reconstructed from the observations y based on the dictionary and sparse coefficients [9], [10]. Often, l_0 or l_1 regularizations are used to enforce the sparsity constraint on the representation coefficients of the signal [11], [12], [13], [14], [15]. The dictionary Ω can be fixed [2], or learned [16], [17], [18], [19]. The declipping problem has been addressed by imposing constraints on the samples in the clipped region. More specifically, the magnitude of the original signal should be greater than or equal to that of the clipped signal in the clipped region. An alternative to the synthesis model is the sparse analysis model, where an analysis dictionary $\Omega \in \mathbb{R}^{p \times m}$ with $p > m$ is used to sparsify the signal $x \in \mathbb{R}^m$ [20], [21], [22], i.e. making Ωx sparse. Only few algorithms, such as the A-SPADE algorithm [23], [24], [25], have considered the use of an analysis model for signal declipping, with a fixed dictionary.

A. Related Work

In [2], the original signal is estimated using the reliable (i.e. unclipped) samples from the observed signal, with the dictionary formed using positions of the unclipped samples. This algorithm does not involve the update of dictionary atoms as they are pre-selected from the unclipped samples of the observed signal. To reconstruct the samples that are clipped, additional constraints are imposed on their amplitudes. This algorithm works better for low clipping levels than for high [2]. When the clipping region is enlarged, the number of reliable (i.e. unclipped) samples that can be used for constructing

B. Li is with China Academy of Space Technology (Xi'an), Xi'an, 710100, China. (E-mail: libin504@hotmail.com)

L. Rencker, M. D. Plumbley, and W. Wang are with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, U.K. (E-mails: [l.rencker, m.plumbley, w.wang]@surrey.ac.uk)

J. Dong is with the College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, Jiangsu, China. (Email: jingdong@njtech.edu.cn)

Y. Luo is with the Department of Data Science, National Physical Laboratory, Teddington, TW11 0LW, U.K. (E-mail: yuhui.luo@npl.co.uk)

The research leading to these results has received funding from the European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet. W. Wang and M. D. Plumbley were also supported by EPSRC grant EP/N014111/1 Making Sense of Sounds. B. Li was supported by the China Scholarship Council. J. Dong was supported by the National Natural Science Foundation of China (61906087), the Natural Science Foundation of Jiangsu Province of China (BK20180692), and the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China (17KJB510025).

the dictionary decreases, and as a result, the quality of the recovered signal deteriorates [2].

In [19], [26], the concept of clipping consistency is exploited to improve the reconstruction of the original signal, where additional constraints are enforced for signal recovery and dictionary learning so that they are “consistent” with the measurement process. For example, the magnitude of the original signal in the clipped region of the observed signal should be greater than the clipping level. Therefore, such information can be used to design consistency constraints to improve signal recovery. However, the algorithm is based on extended linear least-squares, which may be limited for the signals with rapidly changing dynamics (e.g. highly non-stationary signals). It is worth noting that the idea of measurement consistency could also be used to address the problem of signal recovery from other type of nonlinear observations such as quantized or 1-bit samples, as shown in a recent study [27]. Another algorithm called S-SPADE [23], [24] is also based on the synthesis model, which uses l_0 pseudo-norm to measure the sparsity, and l_2 norm to bound the distance between the signal and its sparse approximation.

With a similar heuristic as in the S-SPADE algorithm, the A-SPADE algorithm [23], [24], [25] uses a fixed analysis dictionary such as a discrete Fourier transform (DFT) or a discrete cosine transform (DCT) for iterative signal reconstruction, where the previous estimates are used to obtain the new estimate. There are two main steps in the A-SPADE algorithm. The first step is to obtain the representation matrix using the fixed analysis dictionary and enforce sparsity of the representation matrix. The second step is to apply a constraint on the recovered signal to meet the clipping conditions. The empirical results in [25] show that the A-SPADE algorithm performs better than the S-SPADE algorithm.

Recent research, e.g., [27], has shown that learned dictionaries have the potential to outperform pre-defined dictionaries in representing a signal and providing recovery of the original signal from the clipped signal. Nevertheless, this research has considered only the synthesis model. To the best of our knowledge, dictionary learning with a sparse analysis model has not been applied in the context of signal declipping. In this paper, we will bridge this gap and examine the use of a dictionary learned with a sparse analysis model for signal declipping. This idea is partially inspired by the work [20], where an analysis dictionary was shown to outperform a synthesis dictionary for signal recovery from corrupted observations.

B. Contributions

Our contributions can be summarized as follows. (1) We build a new model (cost function) for signal declipping with the consistency constraint defined on an analysis dictionary. (2) We then develop a two-stage method for optimizing the cost function. This involves dictionary learning using the Analysis SimCO (ASimCO) algorithm [20] in the first stage, and signal recovery in the second stage. To recover the original signal from the clipped signal, we developed two methods based on least squares and projected gradient descent, respectively.

(3) We evaluate the performance of the proposed algorithms with different parameters, in terms of signal to distortion ratio (SDR). We also compare our algorithms with two baseline methods, i.e. A-SPADE [23] and ConsDL [27], and show that our algorithms perform significantly better than the ConsDL algorithm and slightly better than the A-SPADE algorithm.

C. Notations

In this paper, uppercase letters denote matrices, e.g. $Y \in \mathbb{R}^{m \times n}$, and lowercase letters denote vectors, e.g. $y \in \mathbb{R}^m$. The i th row of X is denoted as $X_{i,:}$, and the j th column of X is denoted as $X_{:,j}$. For a matrix X , $\|X\|_0$ denotes the l_0 pseudo-norm, measuring the sparsity, i.e. counting the number of nonzero elements of X . $\|X\|_F$ denotes Frobenius-norm of the matrix X . $\|y\|_1$ denotes the l_1 -norm of the vector y . The symbols \preceq and \succeq denote the element-wise smaller (or equal) and larger (or equal), respectively. I represents an identity matrix, $\mathbf{1}$ is an all-one vector, and $\mathbf{1}$ an all-one matrix. The symbol $(x)_+$ denotes $\max(0, x)$, $(X)_+$ denotes $\max(0, X)$, $(x)_-$ denotes $-(-x)_+$, and $(X)_-$ denotes $-(-X)_+$. The operators $\max(X, Y)$ and $\min(X, Y)$ denote taking the element-wise maximum and minimum of X and Y , respectively. We use C to denote a set. For convenience, we use Ω and Ω^T to represent the analysis and synthesis dictionary, respectively.

D. Organization of the Paper

In Section II, we provide some background materials about the Analysis SimCO algorithm. In Section III, we formulate the declipping problem. In Section IV, we present our method for the declipping problem, including the cost function and the optimization algorithms. In Section V, we present simulation studies of the proposed algorithm, with a focus on the choices of the parameters and comparisons between the least-squares based method and projected gradient descent based method for the recovery of the original signal. In Section VI, we compare our algorithm with the two baseline algorithms. Finally, Section VII draws conclusions and discusses future works.

II. ANALYSIS SIMCO ALGORITHM

In this section, we provide an overview of the ASimCO algorithm that has been already published in [20]. Such materials are not novel, but included to make the algorithms presented later self-contained. In an analysis model, the analysis dictionary $\Omega \in \mathbb{R}^{p \times m}$ is sought to sparsify x [28], [29], [30] as follows

$$a = \Omega x \quad \text{s.t.} \quad \|a\|_0 = p - l \quad (2)$$

where $x \in \mathbb{R}^m$, $a \in \mathbb{R}^p$, $0 \leq l \leq p$, and l is the co-sparsity [31] indicating the number of zero elements in a , and a is the analysis representation vector of the signal x with respect to Ω . In this model, if x is a clean signal, we can obtain the representation vector a directly via multiplying x by the analysis dictionary Ω . Note that the sparsity constraint is an upper bound. In other words, if the value of Ωx is more than l co-sparse, no extra nonzero values will be added to increase

its l_0 pseudo-norm. In practice, Ω can be either a pre-defined fixed analysis dictionary, or an analysis dictionary learned from observed signals. To learn the analysis dictionary, we can use an iterative algorithm alternating between two steps: analysis pursuit and analysis dictionary update. This problem is often called as analysis dictionary learning (ADL) [22], [31], [28], [32], [33]. Given X which contains the training signals, the ADL problem can be described as follows [34]

$$\{\hat{A}, \hat{\Omega}\} = \arg \min_{A, \Omega} \|A - \Omega X\|_F^2 \quad \text{s.t.} \quad \|A_{:,i}\|_0 = p - l, \forall i \quad (3)$$

where $X \in \mathbb{R}^{m \times n}$ is a matrix containing n training examples each of dimension m , and $A \in \mathbb{R}^{p \times n}$ is the analysis representation matrix. In order to mitigate the scale ambiguities, the ASimCO algorithm optimizes the following cost [20], [35], by imposing a constraint on the rows of Ω ,

$$\begin{aligned} \{\hat{A}, \hat{\Omega}\} = \arg \min_{A, \Omega} & \|A - \Omega X\|_F^2 \\ \text{s.t.} & \|A_{:,i}\|_0 = p - l, \forall i \\ & \|\Omega_{j,:}\|_2 = 1, \forall j. \end{aligned} \quad (4)$$

Different from other ADL algorithms, such as Analysis K-SVD [28], multiple atoms in the analysis dictionary Ω can be updated simultaneously in the ASimCO algorithm. The ASimCO algorithm alternates between two stages, i.e. analysis pursuit and analysis dictionary updating, as discussed next.

A. Analysis Pursuit

In this stage, we need to compute the analysis representation matrix A , given the analysis dictionary Ω and the signal X . Similar to (2), we can obtain A as follows

$$A = \Omega X. \quad (5)$$

Since the analysis dictionary is generated arbitrarily, it does not necessarily satisfy the co-sparsity constraints in (2), therefore a hard thresholding operation is applied to ensure the co-sparsity constraint

$$\hat{A} = HT_l(A). \quad (6)$$

This is a nonlinear operation to set the smallest l elements (in magnitude) in each column of A to zeros. Among all the matrices satisfying the co-sparsity constraints, \hat{A} is the best approximation of A in terms of the Frobenius norm error.

B. Dictionary Update

Given X and A , the analysis dictionary Ω can be found by optimizing the following cost function [35], [20]

$$\arg \min_{\Omega} g(\Omega) = \arg \min_{\Omega} \|A - \Omega X\|_F^2 \quad \text{s.t.} \quad \|\Omega_{j,:}\|_2 = 1, \forall j. \quad (7)$$

The unit-norm constraint on Ω suggests that columns of Ω^T sit on the unit sphere $S = \{u \in \mathbb{R}^m : u^T u = 1\}$, i.e. $(\Omega_{j,:})^T \in S$. To optimise this cost function, we can use the gradient descent based method, as detailed in [36] and [20].

The negative gradient of the objective function in (7) with respect to Ω can be obtained as

$$\begin{aligned} H &= -\nabla g(\Omega) \\ &= -\frac{\partial \|A - \Omega X\|_F^2}{\partial \Omega} \\ &= 2AX^T - 2\Omega XX^T. \end{aligned} \quad (8)$$

To update Ω , a line search method accounting for the unit-norm constraint $\|\Omega_{j,:}\|_2 = 1$ was proposed in [20], where the search direction of the j th row of Ω is projected onto the tangent space of S as follows

$$\bar{h}_j = H_{j,:} (I - \Omega_{j,:}^T \Omega_{j,:}) \quad (9)$$

The j th row of Ω can then be updated as follows [20]

$$\hat{\Omega}_{j,:}(\xi) = \begin{cases} \Omega_{j,:} & \text{if } \|\bar{h}_j\|_2 = 0, \\ \Omega_{j,:} \cos(\xi \|\bar{h}_j\|_2) + \left(\frac{\bar{h}_j}{\|\bar{h}_j\|_2} \right) \sin(\xi \|\bar{h}_j\|_2) & \text{otherwise} \end{cases} \quad (10)$$

where ξ is the step size, which can be either fixed, or varied by searching for the optimal value at each iteration in terms of the golden section rule, as detailed in [36], [37]. Readers may refer to [20] for more details about the derivation of (9) and (10) by incorporating $\|\Omega_{j,:}\|_2 = 1$. For the convenience of later use, we abbreviate (10) as follows

$$\hat{\Omega} = w(\Omega). \quad (11)$$

III. MODELLING OF CLIPPED OBSERVATIONS

The clipped signal y in (1) is a distorted version of the original signal x , and can be written as follows

$$y_i = f(x_i) = \begin{cases} \theta^+ & \text{if } x_i \geq \theta^+ \\ \theta^- & \text{if } x_i \leq \theta^- \\ x_i & \text{others} \end{cases} \quad (12)$$

where $\theta^+ > 0$ and $\theta^- < 0$ are positive and negative clipping levels, respectively, as illustrated in Fig. 1. This representation

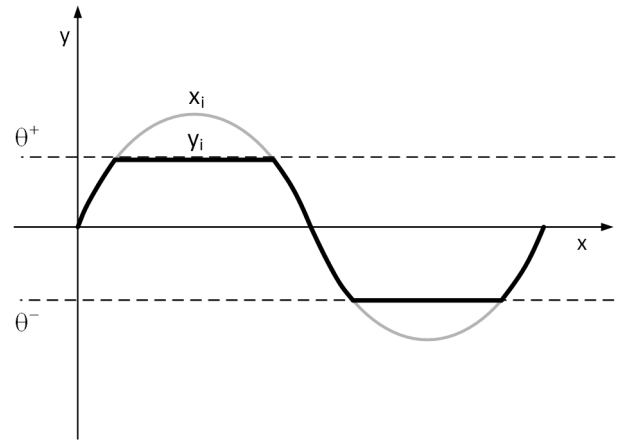


Fig. 1. Hard clipping: y is the observed signal, which contains the nonlinear observations of the original signal x . θ^+ and θ^- are the maximum and the minimum of the observed signal.

can also be written in vector form

$$y = f(x) = M^r x + \theta^+ M^{c+} \mathbf{1} + \theta^- M^{c-} \mathbf{1} \quad (13)$$

where $\mathbf{1}$ is an all-one vector, M^r , M^{c+} and M^{c-} are diagonal binary sensing matrices defining the reliable, positive and negative clipped samples, respectively. In practice, we can select the maximum and minimum value of the observed signal as θ^+ and θ^- , i.e. $\theta^+ = \max(y)$, and $\theta^- = \min(y)$.

Declicking can be regarded as an inverse problem with discarded samples. Given the observed signal y with clipped samples missing, the aim is to reconstruct the original signal x from y , by solving the inverse problem [4], [16], [38], [39].

In declipping problems, the amplitudes of the original signal in the clipping region are greater than or equal to the amplitudes of the observed signal. Therefore, we can define a so-called consistency set, as in [19], in closed form as follows

$$C(y) = \{x | M^r y = M^r x, M^{c+} y \preceq M^{c+} x, M^{c-} y \succeq M^{c-} x\}. \quad (14)$$

This set contains all the possible x that could have generated the observation y that is *consistent* with the observation (i.e. measurement) process, which, in this case, is described by the clipping function.

IV. PROPOSED METHOD

A. Cost Function

Our method is built on the following cost function

$$\begin{aligned} \{\hat{\Omega}, \hat{X}, \hat{A}\} = \arg \min_{\Omega, X, A} \|A - \Omega X\|_F^2 \\ \text{s.t. } \|A_{:,i}\|_0 = p - l, \forall i \\ \|\Omega_{j,:}\|_2 = 1, \forall j \\ X \in C(Y) \end{aligned} \quad (15)$$

where $X \in \mathbb{R}^{m \times n}$ represents the signal to be estimated, $Y \in \mathbb{R}^{m \times n}$ represents the clipped signal, and $C(Y)$ is a consistency set defined on matrices, i.e. replacing y by Y and x by X in (14). The cost function (15) consists of two parts. The first part, $\|A - \Omega X\|_F^2$, aims to learn the analysis dictionary Ω that can sparsify the signal X . The constraint on $\|A_{:,i}\|_0$ is used to enforce the sparsity of the representation coefficients A , and the constraint on $\|\Omega_{j,:}\|_2$ is applied to mitigate the scale ambiguity of the analysis dictionary Ω . The second part takes into account the nonlinear observations from the clipped signal Y , and enforces the recovered signal X to be on the set $C(Y)$. In practice, Y could be formed by splitting the clipped signal into n segments each of dimension m . Our aim is to learn the analysis dictionary Ω and estimate the signal X , given the observed signal Y .

B. Optimization Process

To optimize the cost function (15), we develop a two-stage method where the unknown variables are estimated in an alternating manner, as commonly adopted in sparse representations and analysis dictionary learning.

In the first stage, given $X^{[t]}$ obtained in iteration t , $\Omega^{[t+1]}$ and $A^{[t+1]}$ are updated in the $(t+1)$ -th iteration, based on the following cost function

$$\begin{aligned} \{\Omega^{[t+1]}, A^{[t+1]}\} = \arg \min_{\Omega, A} \|A - \Omega X^{[t]}\|_F^2 \\ \text{s.t. } \|A_{:,i}\|_0 = p - l, \forall i \\ \|\Omega_{j,:}\|_2 = 1, \forall j. \end{aligned} \quad (16)$$

Note that $X^{[0]}$ is initialized as Y . This stage can be achieved by the ASimCO algorithm [20] as reviewed in Section II.

The second stage is to find X , given Ω and A , under the constraint on X to enforce consistency, i.e. $X \in C(Y)$, as follows

$$\begin{aligned} \hat{X} = \arg \min_X \|A - \Omega X\|_F^2 \\ \text{s.t. } X \in C(Y). \end{aligned} \quad (17)$$

The proposed sparse analysis model based declipping (SAD) algorithm can be outlined in Algorithm 1.

Algorithm 1: Proposed SAD

1. **Initialization:** $\Omega^{[0]}, X^{[0]} = Y, t = 0, \mu$
 2. **While** not converged **do**
 3. Update Ω and A by solving (16) using ASimCO.
 4. Update X by solving (17).
 5. $t = t + 1$
 6. **End while**
-

To optimise the cost function (17), we present two methods, i.e. the least squares based method and projected gradient based method, respectively.

The least squares based method can be achieved by approximating the constrained optimization problem (17) as an unconstrained one. We note that the term $\|M^r(Y - X)\|_F^2$ promotes the recovered signal to be close to the original signal for the unclipped part of the signal, and the terms $\|M^{c+}(\theta^+ \mathbf{1} - X)_+\|_F^2$ and $\|M^{c-}(\theta^- \mathbf{1} - X)_-\|_F^2$ promote the amplitudes of the recovered samples in the clipped regions to be consistent with the clipping function, where $\mathbf{1} \in \mathbb{R}^{m \times n}$ is an all-one matrix. As a result, instead of optimizing (17), we can optimise the following cost function

$$\begin{aligned} \hat{X} = \arg \min_X \left\{ \|A - \Omega X\|_F^2 + \mu \left[\|M^r(Y - X)\|_F^2 \right. \right. \\ \left. \left. + \|M^{c+}(\theta^+ \mathbf{1} - X)_+\|_F^2 + \|M^{c-}(\theta^- \mathbf{1} - X)_-\|_F^2 \right] \right\}. \end{aligned} \quad (18)$$

where a regularization term controlled by a parameter μ is used to approximate the constraint $X \in C(Y)$. A relaxed problem similar to the one proposed in (18) has been used in [4], [27], [40], [41] in the context of the synthesis model with a synthesis dictionary [42], [43], [44].

We then introduce an auxiliary function $Z(X) \in \mathbb{R}^{m \times n}$ defined as follows

$$Z(X) = M^r Y + M^{c+} \max(\theta^+ \mathbf{1}, X) + M^{c-} \min(\theta^- \mathbf{1}, X) \quad (19)$$

With $Z(X)$, we can simplify (18) to

$$\hat{X} = \arg \min_X \|A - \Omega X\|_F^2 + \mu \|Z(X) - X\|_F^2 \quad (20)$$

With a least squares based method [45], [46], [47], the solution of X can be obtained analytically from (20) as

$$X^{[t+1]} = (\Omega^T \Omega + \mu I)^{-1} (\Omega^T A + \mu Z(X^{[t]})) \quad (21)$$

where $I \in \mathbb{R}^{m \times m}$ is an identity matrix, A and Ω are obtained by following the dictionary update step in (16). In our work, Ω is real-valued, and $\Omega^T \Omega$ is a positive definite matrix. As a result $(\Omega^T \Omega + \mu I)$ is a full-rank invertible positive definite matrix. Empirically, we observed that the matrix inversion was stable in our experiments. Note that the use of the auxiliary function $Z(X)$ is to facilitate the solution of (20). Although the three types of samples are regularized with a single μ in (18), there may be a potential scope to treat them differently with different regularization parameters.

The second method for optimising (17) is based on the projected gradient algorithms [47], [48]. The gradient of the objective function (17) with respect to X is

$$\begin{aligned} \nabla \|A - \Omega X\|_F^2 &= \frac{\partial \|A - \Omega X\|_F^2}{\partial X} \\ &= -2\Omega^T A + 2\Omega^T \Omega X. \end{aligned} \quad (22)$$

Thus, $X^{[t+1]}$ can be obtained as

$$X^{[t+1]} = X^{[t]} - \lambda(-2\Omega^T A + 2\Omega^T \Omega X^{[t]}) \quad (23)$$

where λ is a step size. A projection step is then used to enforce $X^{[t+1]} \in C(Y)$ as follows

$$\begin{aligned} X^{[t+1]} \leftarrow M^r Y + M^{c+} \max(\theta^+ \underline{1}, X^{[t+1]}) \\ + M^{c-} \min(\theta^- \underline{1}, X^{[t+1]}) \end{aligned} \quad (24)$$

where the measurement matrices are the same as those in Equation (14), and \leftarrow means updating $X^{[t+1]}$ with the formula on the right hand side.

With the SAD algorithm, we can finally obtain the analysis dictionary Ω , the analysis representation matrix A and the estimated signal X . Algorithm 2 summarizes the implementation details of the SAD algorithm.

Algorithm 2: Implementation details of SAD

1. **Initialization:** $\Omega^{[0]}, X^{[0]} = Y, t = 0, l, \mu, \lambda$
 2. **While** not converged **do**
 3. **Input:** $X^{[t]}, \Omega^{[t]}, \Omega_{[1]} = \Omega^{[t]}, k = 0, g(\Omega_{[0]}) = 0$
 - a. **do**
 - b. $\Omega_{[k+1]} = w(\Omega_{[k]})$
using equation (11) for function $w(\Omega)$
 - c. $A_{[k+1]} = HT_l(\Omega_{[k+1]} X^{[t]})$
using equation (6) for function $HT_l(\Omega X)$
 - d. $k = k + 1$
 - e. **until** a pre-defined number of iterations is reached.
 - f. **output:** $\Omega^{[t+1]} = \Omega_{[k+1]}, A^{[t+1]} = A_{[k+1]}, Z^{[t+1]}$
 4. $\Omega = \Omega^{[t+1]}, A = A^{[t+1]}, Z^{[t+1]}$
 - a. using equation (21) or (23) (with (24)) to get $X^{[t+1]}$
 - b. **output:** $X^{[t+1]}$
 5. $t = t + 1$
 6. **End while**
 7. **Output:** X
-

C. Algorithm Convergence and Initialization

The SAD algorithm alternates among the steps of analysis sparse coding, analysis dictionary update and signal estimation. The algorithms for estimating X , i.e. the operation 4.a in Algorithm 2 have been shown to be convergent in [20] and [49]. However, convergence of Step 3 cannot be ensured. We observed that $g(\Omega_{[k]})$ may increase occasionally, which could be caused by the update of X with (20) and (19) in which the comparison between X and $\theta^+ \underline{1}$ or $\theta^- \underline{1}$ is performed. However, the decreasing trend of $g(\Omega_{[k]})$ is often restored automatically with further iterations. In practice, we did not observe any other issues about the convergence of the algorithm. We have used a pre-defined number of iterations (e.g. 10) as the stopping criterion for the inner loop for updating the analysis dictionary. In Algorithm 2, we use the superscript t to denote the iteration index for the outer loop, while the subscript k for the inner loop.

In the SAD algorithm, the initial dictionaries Ω can be set as random matrices with normalized rows, or as pre-defined dictionaries such as the DCT dictionary. The initial dictionaries can affect the results, as observed in our experiments. We will examine the impact of the initial analysis dictionary on the declipping performance in Section V.

V. SIMULATIONS

In this section, we focus on demonstrating the performance of the proposed SAD algorithm using different parameters, on sound datasets. Our algorithms can be easily applied to other data, such as images. We use SDR as the performance metric, which is defined as follows

$$\text{SDR}(\hat{x}, x) = 20 \log \frac{\|x\|_2}{\|x - \hat{x}\|_2} \quad (25)$$

where \hat{x} is the estimated signal, and x is the original signal.

In the following subsections, we will show the setup of our experiments, the performance of the SAD algorithm, including the influence of the parameters m and l , and the initialization of Ω on the performance of the algorithm. The code of our proposed SAD algorithm can be downloaded from Github¹. We will present the comparisons of our algorithms with the baseline algorithms, i.e. ConsDL [19] and A-SPADE [23], in Section VI.

A. Experimental Setup

To form Y , we take the clipped signal and divide it into n segments, each of length m , with overlaps between the adjacent segments, using a sliding window. For example, if the clipped signal has 16384 samples, we can form Y as a matrix 64×16321 , using a sliding window of $m = 64$ samples with a hop size equal to one sample. In the experiments in this section, we tested several different values for m , i.e. $m = 16, 32, 64$, and 128. The specific values used are discussed in the corresponding subsections.

In our experiments, the clipping level θ was set using a routine from [27], which takes an input SDR and outputs the

¹http://github.com/BinLi504/SAD_matlabopen

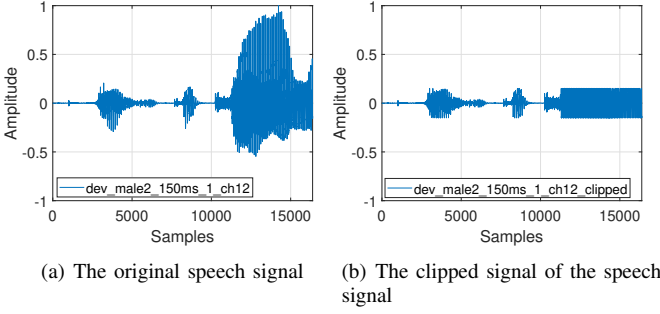


Fig. 2. The waveform of the original speech signal and the clipped signal used for performance tests. The original signal was normalized to $[-1, 1]$, before being clipped at the level $\theta = 0.15$.

clipping level. The clean signal was normalized before it was clipped at the specified level. More details about the generation of clipped signals are given in Section VI-A.

Our experiments were performed using Matlab on the Intel Core i5-7500 CPU, with memory 8 G-Bytes, on Windows 10 operating system.

B. Empirical Tests of Parameters

We first test the performance of the proposed algorithm (with the least-squares based update for X) using a speech signal “dev_male2_150ms_1_ch12.wav” from the SISEC dataset [50]. These signals contain stereo recordings, with 8 bits per channel, sampled at 16 kHz. One of the channels is used as test data. The original signal is relatively long, therefore we cut it into 16384 samples, as shown in Fig. 2 (a). We study empirically the set up of important parameters such as co-sparsity and dimension of dictionary atoms. Here we use a relatively short signal for parameter tuning for the proposed SAD method, however, we have also used longer and more test signals in Section VI for performance comparisons on speech and music datasets. The clipping level θ is set as 0.15. The waveform of the clipped signal is shown in Fig. 2 (b). We set $m = 64$, $l = 120$, and the number of iterations as 20000.

1) *Selection of Co-sparsity l and m :* In the SAD algorithm, the observed signal y is divided into segments each of length m (i.e. equal to the dimension of dictionary atoms in terms of our formulations discussed in Section IV), and the estimated signal is finally obtained by concatenating the segments with an overlap-and-add technique. The dimension of dictionary atoms m and the selection of co-sparsity l affect the quality of the recovered signal. We study the impact of the co-sparsity l on the performance of the proposed algorithm. The value of l is varied, with a step increase of 8. The clipping level is fixed as $\theta = 0.15$.

Fig. 3 shows the results with different l 's, when $m = 128$, 64, 32, and 16, respectively. It can be seen that the setting of l is related to m . For example, when $m = 128$, the best choice for l is 232, while for $m = 64$, the best choice for l is 120. When $m = 32$, the best choice for l is 56, while for $m = 16$, the best choice for l is 24. When we use different m 's, l should be set accordingly. For each m value, we choose l empirically in terms of the reconstruction quality. The SDR results with

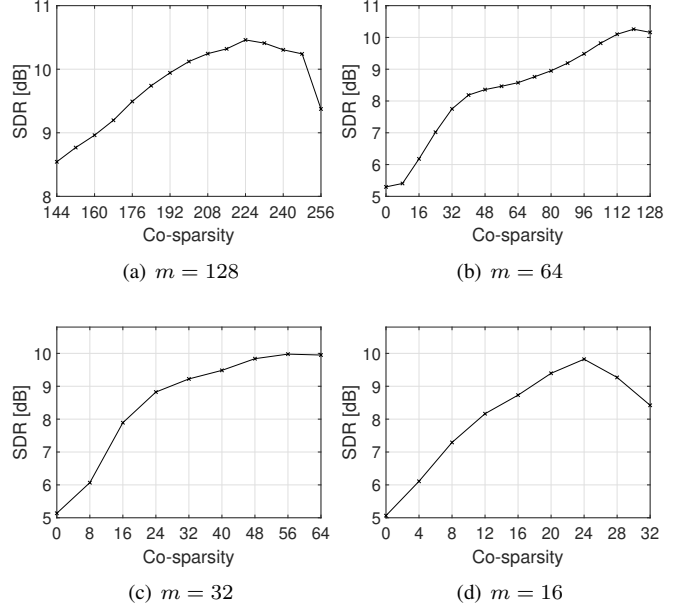


Fig. 3. The SDR achieved using different co-sparsity l . The setting of l is related to the dimension of dictionary atoms m . For a given m , the choice l can be found empirically.

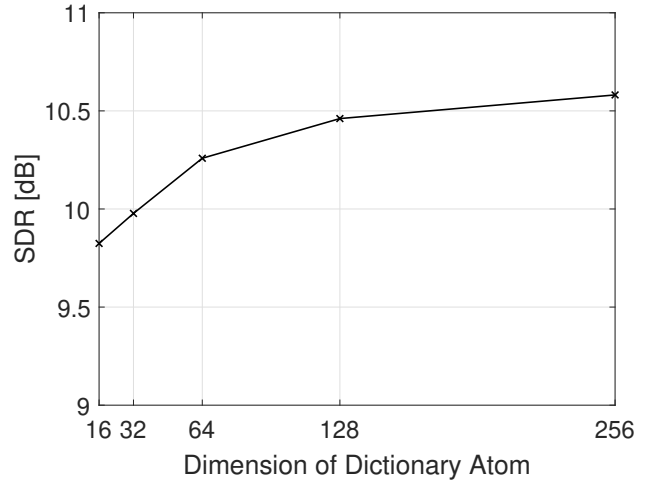


Fig. 4. The SDR of the recovered signal, achieved with different dimension of dictionary atoms m . The signal reconstruction performance by the proposed algorithm increases with the increase of m , however, the performance increase tends to be saturating with larger m .

different m 's are presented in Fig. 4. It can be observed that the SDR increases with the increase of m , but such improvement is flattened when m is relatively large. Although only a single signal was used in this test, we observed empirically that the performance trend is similar when other signals were used. Thus, the parameters tuned in this way seem to be appropriate for subsequent experiments.

2) *Signal Recovery:* To illustrate the performance of the proposed algorithm, we show the plots of three segments of the recovered signal and the original signal in Fig. 5. Here $m = 64$ and $l = 120$. It can be seen that the recovered signal is similar to the original signal.

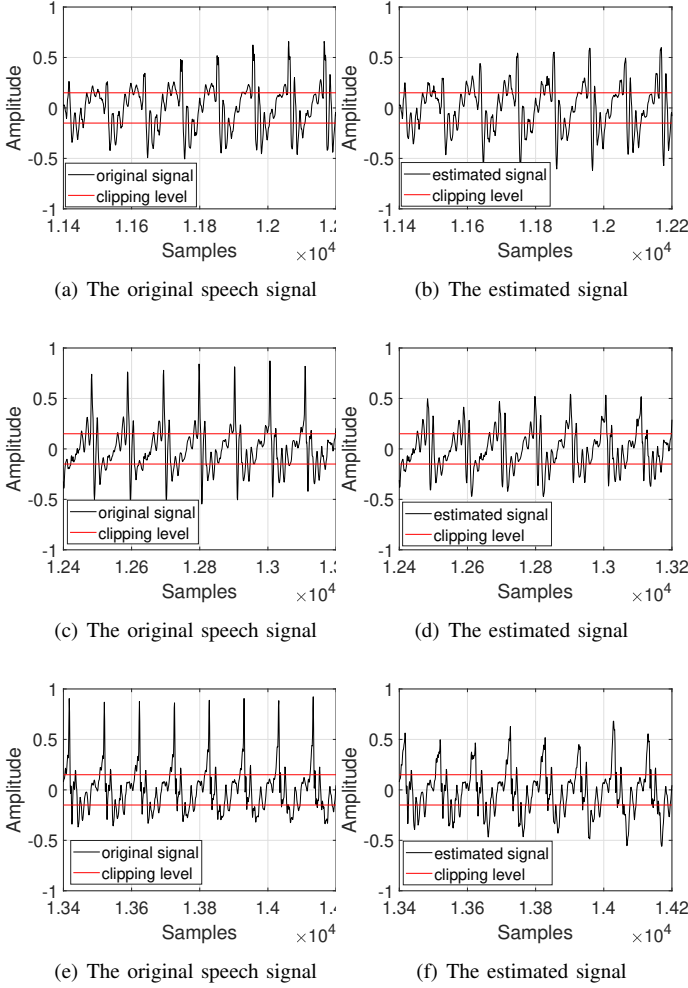


Fig. 5. Signal recovery examples: (a) (c) and (e) show three segments of an original clean speech signal and the clipping level. (b), (d) and (f) show the corresponding segments of the estimated (i.e. declipped) signal.

C. Dictionary Initialization: Random versus DCT

With the speech signal used in the previous section, we compare the performance difference for the dictionaries initialized as a random Gaussian matrix and a uniform distribution matrix. We run the experiments ten times and observe the effect of these initial dictionaries on the signal recovery results. The estimate of X is based on the least-squares update. The average results of these tests are presented in Fig. 6. It can be seen that the results are similar. The standard deviation of the SDR results in the 10 tests with Gaussian matrix is 0.083 dB and the uniform matrix is 0.052 dB. It appears that the randomly initialised dictionaries with these two distributions give similar SDR results.

We also compare the randomly initialised analysis dictionary with the dictionary initialised as DCT. As the two random dictionaries tested above give similar results, we initialise the random matrix by generating the elements following a Gaussian distribution with zero mean and unit variance. The atoms in the initial dictionaries were normalized. In this experiment, we have increased the number of test signals to

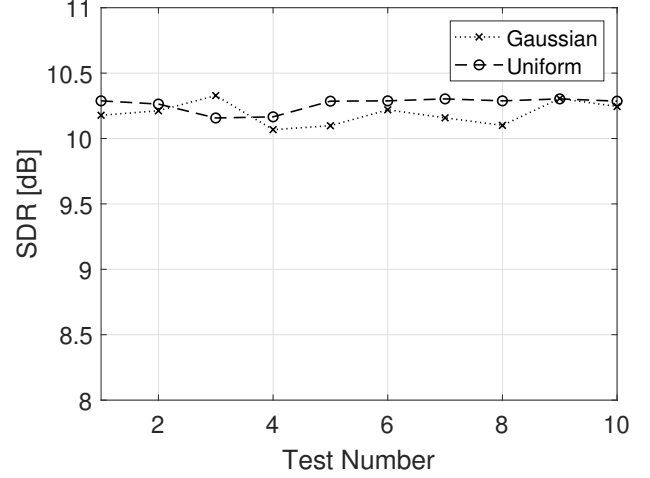


Fig. 6. The SDR of the declipped signal, achieved with the analysis dictionary initialized as a random matrix whose elements follow Gaussian or uniform distributions with zero mean and unit variance.

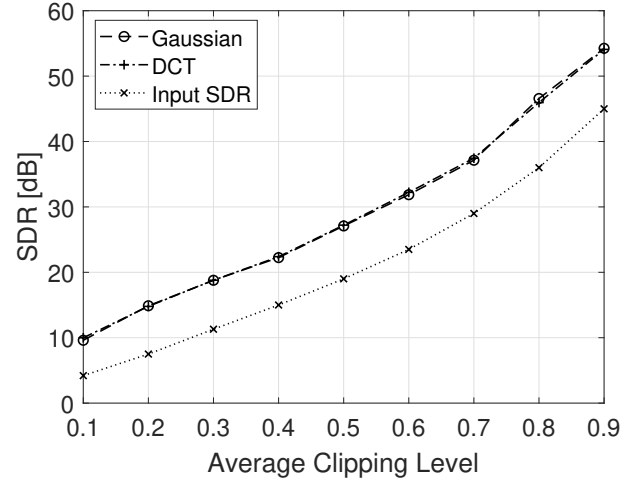


Fig. 7. The SDR of the declipped signal, achieved either with the dictionary initialized as a random matrix whose elements follow Gaussian distribution or with a DCT dictionary.

obtain more reliable results. We used 10 speech signals², and the length of each signal is about 5 to 6 seconds, including about 80000 to 96000 samples (as the sampling rate is 16 kHz). The parameters of the SAD algorithm are set as $m = 72$ and $l = 136$, and X is solved with the least-squares update.

Fig. 7 shows the SDR of the recovered signal with the dictionary learning algorithm in which the dictionary is initialised as either a random Gaussian matrix or as a DCT dictionary. Overall, using the DCT dictionary performs similarly to using a Gaussian dictionary in the initialization. When the clipping level is at 0.1, the SDR obtained by using DCT is 0.3 dB higher than that using the Gaussian dictionary.

²<http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/Audio/Main/>

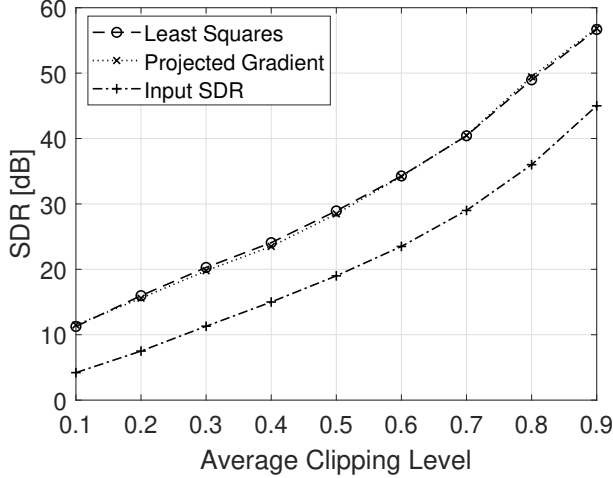


Fig. 8. The SDR of the declipped signal, obtained with the proposed SAD algorithm, where the least squares method and the projected gradient method are used for updating X , respectively.

D. Least Squares versus the Projected Gradient

In the SAD algorithm, two methods can be used to obtain X , i.e. the least squares based method and the projected gradient based method. In this section, we compare the performance of these two methods. We use the same experimental set up as in Section V-C, with the analysis dictionary initialised with the DCT dictionary. Fig. 8 shows that the SDRs obtained by the two methods are very similar.

We have tuned the parameters such as the step size λ , by running a set of experiments. In particular, we tested different combinations of the parameters $\theta = 0.2$, and 0.8 , $\lambda = 0.1$, 0.25 , and 0.4 , and the number of iterations equal to 1, 10 and 20, respectively. The test results are given in Table I. The results show that the step size and the number of iterations can affect the SDR results of the recovered signals. Based on these tests, we chose the step size at 0.25 and the number of iterations at 10 in the subsequent experiments.

We also analyzed the impact of μ on the results by changing μ from 0.4 to 1.8, while keeping other parameters unchanged. The test results for the clipping level at 0.15 (i.e. input SDR at 5 dB) are given in Table II. From the table, we can see that the selection of μ has a very small influence on the results.

These two methods seem to offer very similar performance, and both need to tune a parameter (i.e. μ in the least squares based method, and λ in the projected gradient method). However, we found empirically that their computational complexities can be different. The projected gradient based method contains an additional loop (i.e. the inner loop for updating X), and if multiple iterations in the inner loop are performed, its complexity will become higher than the least squares based method. For this reason, we will only use the least-squares based method for the comparisons in the next section.

VI. COMPARISON WITH BASELINE ALGORITHMS

In this section, we compare the performance of our proposed SAD algorithm with two recent baseline algorithms, i.e. the ConsDL algorithm [19] and the A-SPADE algorithm [23].

TABLE I
TESTS FOR DIFFERENT COMBINATIONS OF PARAMETERS (CLIPPING LEVEL, STEP SIZES AND ITERATION NUMBERS) IN THE PROPOSED METHOD WITH PROJECTED GRADIENT.

θ	λ	iterations	SDR (dB)
0.2	0.1	1	15.36
0.2	0.1	10	14.70
0.2	0.1	20	14.76
0.2	0.25	1	15.35
0.2	0.25	10	14.54
0.2	0.25	20	14.54
0.2	0.4	1	13.47
0.2	0.4	10	12.10
0.2	0.4	20	12.01
0.8	0.1	1	39.88
0.8	0.1	10	39.90
0.8	0.1	20	39.98
0.8	0.25	1	39.94
0.8	0.25	10	40.91
0.8	0.25	20	41.12
0.8	0.4	1	42.32
0.8	0.4	10	41.20
0.8	0.4	20	41.31

TABLE II
THE INFLUENCE OF DIFFERENT μ VALUES ON SIGNAL RECOVERY PERFORMANCE IN TERMS OF SDR OF THE RECOVERED SIGNAL.

μ	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
SDR	8.64	8.64	8.66	8.66	8.66	8.66	8.66	8.64

A. Datasets and Test Preparation

Both speech and music data were used in our evaluations. The speech signals were taken from the Trunk dataset. We used four folders from this dataset, which are respectively, AT-20130718-lws, Adminvox-05232006, 1snoke-20120412-hge, and 1337ad-20170321-ajg. In total, there are 50 speech signals from these four folders, sampled at 16 kHz. The length of each signal is about 4 to 7 seconds. The music signals were taken from the DSD100 dataset³. The music files used are the vocals of 055 – Angels In Amplifiers - I’m Alright, vocals of 081 - Patrick Talbot - Set Me Free and vocals of 049 - Young Griffio - Facade, downloaded from DSD100 dataset. Each of these files contains sound recordings of approximately 45 to 49 seconds, sampled at 44.1 kHz. We then pre-processed the data, and generated 50 signals each of 1.9 seconds, i.e. 81920 samples.

The clipped signals were generated in the same way for speech and music signals, by following [27]. More specifically, we set the same input SDR for each of the 50 signals, and tune the level of clipping to match the input SDR. We also tune the input SDR so that the level of clipping is achieved approximately at the level specified. We then calculate the average level of clipping over these 50 signals. For example, when the input SDR is 4.2 dB for 50 speech signals, the average θ is approximately 0.1. We varied the clipping level θ from 0.1 to 0.9, which corresponds to the input SDR from

³<https://www.loria.fr/~aliutkus/DSD100subset.zip>

TABLE III
MAPPING BETWEEN AVERAGE CLIPPING LEVEL θ AND INPUT SDR OF THE
SPEECH SIGNALS (S/S), AND MUSIC SIGNALS (S/M) USED IN THE
EXPERIMENTS

θ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
S/S	4.2	7.5	11.3	15	19	23.5	29	36	45
S/M	3.5	7	10.8	14	18	22	27	33	42

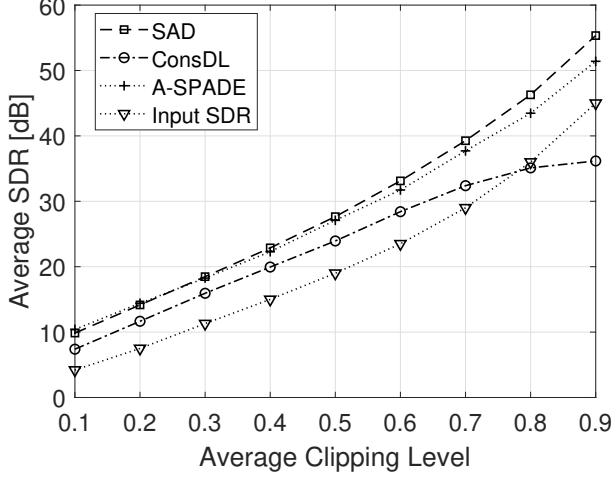


Fig. 9. The average SDR over all the recovered speech signals obtained by the SAD algorithm as compared with A-SPADE and ConsDL, for different clipping levels.

4.2 dB to 45 dB, as shown in the second row of Table III. Note that for the same input SDR, different signals may lead to slightly different θ values. Therefore, the θ is an average θ for all the clipped signals used in our tests, tuned to be approximately at the corresponding clipping level.

The implementation of the ConsDL algorithm [19] comes from the author's personal homepage⁴. For the A-SPADE algorithm [23] we used its version 1.0, downloaded from the author's homepage⁵. In our experiments, the default settings of the ConsDL (i.e. the size of overlapping time frames $N = 256$ with rectangular windows and 75% overlap) and A-SPADE algorithms were used. The parameters used in the SAD algorithm are set as $p = 144$, $m = 72$ and $l = 120$, with initial analysis dictionary Ω set as a DCT dictionary matrix. The condition for stopping algorithm iterations is that the difference between the values of the cost function at two consecutive iterations is less than 0.001. The same experimental setup has been used for the experiments on speech and music data. We take the average of the SDRs calculated from the recovered signals from the 50 tests as the performance metric.

B. Results on Speech Signals

Fig. 9 shows the average SDR of all the recovered speech signals for each clipping level. From this figure, we can see

⁴https://www.cvssp.org/Personal/LucasRencker/software.html#DL_for_declipping

⁵<http://spade.inria.fr>

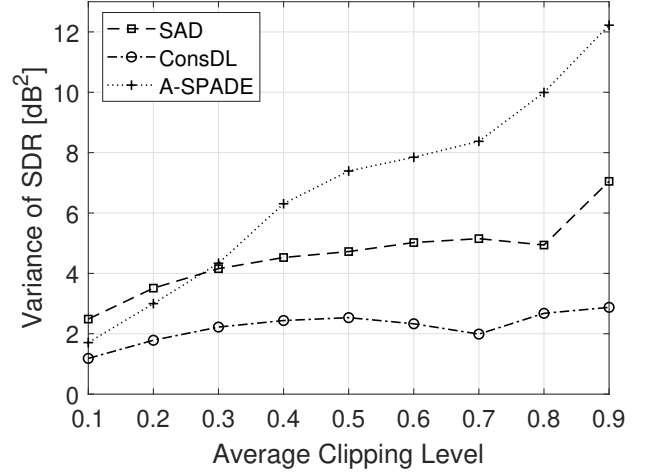


Fig. 10. The variance of the SDRs for all the recovered speech signals obtained by the SAD as compared with the A-SPADE and ConsDL algorithms, for each clipping level.

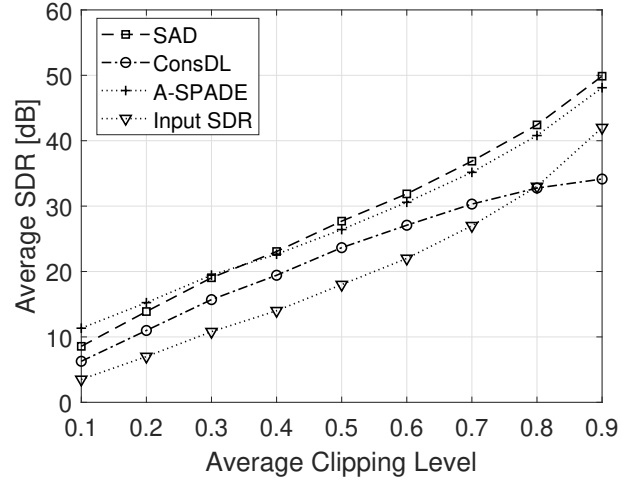


Fig. 11. The average SDR of all the recovered music signals obtained by the proposed SAD algorithm, as compared with the A-SPADE and ConsDL algorithms for different clipping levels. For comparison, the average input SDR is also shown for each clipping level.

that the proposed SAD algorithm performs better than other algorithms, except when $\theta = 0.1$, and 0.2 , in which case, A-SPADE performs better. Fig. 10 shows the variance of the SDR results for all the tests on the 50 speech signals obtained by the SAD for each clipping level, as compared with the baseline algorithms. It can be observed that the ConsDL algorithm gives the smallest variance, while the variance of A-SPADE is higher than those of other compared methods including the proposed SAD algorithms, except when $\theta < 0.2$. It is interesting to note that the variance of all the algorithms increases with the increase in θ , which is probably not surprising considering the fact that the average SDR improvements also increase with the increase in θ .

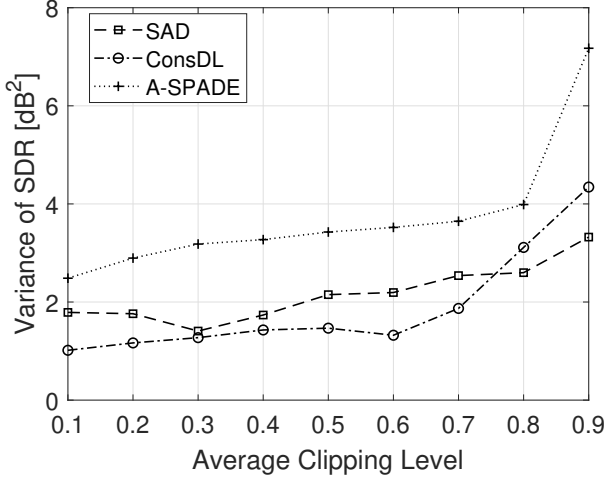


Fig. 12. The variance of the SDR results for all the recovered music signals obtained by the SAD, as compared with the A-SPADE and ConsDL algorithms for each clipping level.

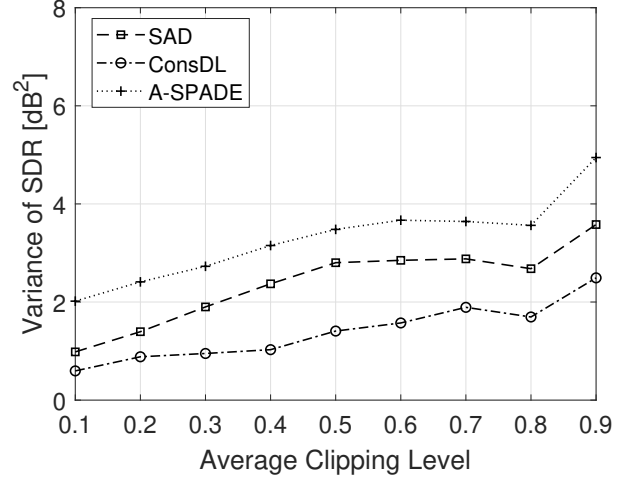


Fig. 14. The variance of the SDR results for all the tests on the recovery of the down-sampled music signals, obtained by the SAD, as compared with the A-SPADE and ConsDL algorithms for different clipping levels.

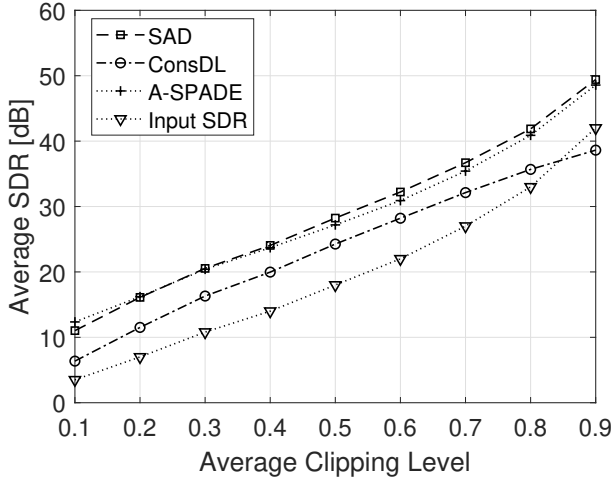


Fig. 13. The average of the SDR results for all the tests on the recovery of the down-sampled music signals, obtained by the SAD, as compared with the A-SPADE and ConsDL algorithms for different clipping levels. For comparison, the average input SDR is also shown for each clipping level.

C. Results on Music Signals

Fig. 11 shows the average SDR results of all the tests for the music signals. The proposed method performs better than ConsDL for all the clipping levels, and slightly better than A-SPADE, except for low clipping levels. Fig. 12 shows the variance of the SDR results of the recovered music signals, for each clipping level. From this figure, we can see that the variance of the SAD algorithm is smaller than that of the A-SPADE algorithm but larger than that of the ConsDL algorithm.

In the A-SPADE algorithm, we used the default parameter set up in its original codes, which was for signals sampled at 16 kHz, and may not be optimal for the experiments on the music signals sampled at 44.1 kHz. In practice, it may not be a trivial task to tune the parameters of A-SPADE to obtain the optimal performance for a different sampling rate. To address

this issue, we run another set of experiments, where we first re-sample music signals from 44.1 kHz to 16 kHz. We then run the tests of all the compared algorithms on the re-sampled music signals. Fig. 13 shows the average SDR of the recovered 50 music signals obtained by the proposed SAD algorithm, as compared with A-SPADE and ConsDL. From this figure, we can see that the SDR results obtained are quite similar to those in Fig. 11. This shows that the influence of sampling rate on the performance of the algorithms compared is very small. Fig. 14 shows the variance of the SDR results for all the tests on the re-sampled music signals obtained by the SAD for each clipping level, as compared with the baseline algorithms. From the figure, we can see that the general trend of performance is similar to those shown in Fig. 12, although it appears to be more stable with respect to different clipping levels. The ConsDL algorithm still gives the smallest variance. The variance of A-SPADE is higher than those of other compared methods including the proposed SAD algorithm.

D. A Note on Running Speed

We found empirically that the proposed SAD algorithm is computationally less efficient as compared with A-SPADE and ConsDL. To process a signal of 81920 samples with a dictionary of dimension 144×64 , the proposed SAD algorithm using least-squares based update for X took about 18 minutes to run for 500 (outer) iterations, with the simulation environment we mentioned earlier. Using the projected gradient based method, the proposed SAD algorithm needs to run an additional loop (i.e. the inner loop for updating X). With the same number of iterations for the outer loop, i.e. 500, the SAD algorithm took about 17.5 and 26.5 minutes, respectively, when the inner loop was run for one iteration and 10 iterations, respectively. However, the A-SPADE and ConsDL algorithms took about 6 and 6.5 minutes, respectively.

VII. CONCLUSION

We have presented a sparse analysis-model based signal declipping algorithm. In our proposed SAD algorithm, the cost function is established on a sparse analysis model, and the clipped signal is restored by using the analysis dictionary learned directly from the observed signal, via the Analysis SimCO algorithm. The evaluation results showed that our proposed algorithm offers better performance in terms of SDR and more stable results in terms of variance, than two recent baselines, namely, the ConsDL and A-SPADE algorithms.

In our work, we considered only real-valued analysis dictionary Ω . The method could potentially be extended to complex-valued dictionaries, such as the dictionaries learned with complex non-negative matrix factorization [51], which uses another matrix to encode the phase term of the complex numbers. Including phase term could help maintain phase coherence between different frequency components within the signal [51].

In addition, we used a fixed step size in the projected gradient method, which could be replaced by a variable step size, and optimized using e.g. the golden section rule, as used in [36] and [37]. Other possible directions of future work include extension of the SAD algorithm for noisy signal declipping, the employment of the dictionary pair learning (DPL) [52] for declipping, and comparisons with other analysis dictionary learning algorithms such as [53].

ACKNOWLEDGEMENT

The authors wish to thank the reviewers and the editor for their constructive comments that have helped improve this article.

REFERENCES

- [1] P. Závika, P. Rajmic, A. Ozerov, and L. Rencker, “A survey and an extensive evaluation of popular audio declipping methods,” *IEEE Journal of Selected Topics in Signal Processing*, Dec. 2020. DOI: 10.1109/JSTSP.2020.3042071.
- [2] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “A constrained matching pursuit approach to audio declipping,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 329–332.
- [3] S. Foucart and J. Li, “Sparse recovery from inaccurate saturated measurements,” *Acta Applicandae Mathematicae*, vol. 158, no. 1, pp. 49–66, 2018.
- [4] J. N. Laska, P. T. Boufounos, M. A. Davenport, and R. G. Baraniuk, “Democracy in action: Quantization, saturation, and compressive sensing,” *Applied and Computational Harmonic Analysis*, vol. 31, pp. 429–443, Nov. 2011. DOI: 10.1016/j.acha.2011.02.002.
- [5] H. Mansour, R. Saab, P. Nasiopoulos, and R. Ward, “Color image desaturation using sparse reconstruction,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 778–781.
- [6] T. Blumensath and M. E. Davies, “Iterative thresholding for sparse approximations,” *Journal of Fourier Analysis & Applications*, vol. 14, no. 5, pp. 629–654, 2008.
- [7] D. L. Donoho and M. Elad, “Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ_1 minimization,” *Proceedings of the National Academy of Science*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [8] C. Bilen, A. Ozerov, and P. Pérez, “Audio declipping via nonnegative matrix factorization,” pp. 1–5, 2015. DOI: 10.1109/WASPAA.2015.7336948.
- [9] S. S. Chen and D. M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [10] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.
- [11] M. Harvilla and R. Stern, “Least squares signal declipping for robust speech recognition,” *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2073–2077, Jan. 2014.
- [12] A. Ozerov, C. Bilen, and P. Perez, “Multichannel audio declipping,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 659–663.
- [13] C. Gaultier, S. Kitić, R. Gribonval, and N. Bertin, “Sparsity-based audio declipping methods: Overview, new algorithms, and large-scale evaluation,” May 2020. [Online]. Available: <https://arxiv.org/abs/2005.10228>.
- [14] F. Lieb and H. G. Stark, “Audio inpainting: Evaluation of time-frequency representations and structured sparsity approaches,” *Signal Processing*, vol. 153(DEC), pp. 291–299, 2018.
- [15] I. Touni and V. Emiya, “Sparse non-local similarity modeling for audio inpainting,” Apr. 2018, pp. 576–580. DOI: 10.1109/ICASSP.2018.8462187.
- [16] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “Audio inpainting,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 922–932, Mar. 2012. DOI: 10.1109/TASL.2011.2168211.
- [17] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [18] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” *Inverse Problems*, vol. 23, no. 3, p. 947, 2007.
- [19] L. Rencker, F. Bach, W. Wang, and M. D. Plumbley, “Consistent dictionary learning for signal declipping,” in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2018, pp. 446–455.
- [20] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Z.-F. Han, and J. Chambers, “Analysis SimCO algorithms for sparse analysis model based dictionary learning,” *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 417–431, 2015.
- [21] J. Dong, Z. Han, Y. Zhao, W. Wang, A. Prochazka, and J. Chambers, “Sparse analysis model based multiplicative noise removal with enhanced regularization,” *Signal Processing*, vol. 137, pp. 160–176, 2017.
- [22] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, “Constrained overcomplete analysis operator learning for cosparse signal modelling,” *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2341–2355, 2013.
- [23] S. Kitić, N. Bertin, and R. Gribonval, “Sparsity and cosparsity for audio declipping: A flexible non-convex approach,” in *Latent Variable Analysis and Signal Separation*, E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský, Eds., Cham: Springer International Publishing, 2015, pp. 243–250.
- [24] P. Závika, P. Rajmic, O. Mokřý, and Z. Průša, “A proper version of synthesis-based sparse audio declipper,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 591–595.
- [25] P. Závika, P. Rajmic, Z. Průša, and V. Veselý, “Revisiting synthesis model in sparse audio declipper,” in *Latent Variable Analysis and Signal Separation*, Y. Deville, S. Gannot, R. Mason, M. D. Plumbley, and D. Ward, Eds., Cham: Springer International Publishing, 2018, pp. 429–445.

- [26] A. J. Weistein and M. B. Wakin, "Recovering a clipped signal in sparseland," *Sampling Theory in Signal and Image Processing*, vol. 12, no. 1, pp. 55–69, 2013.
- [27] L. Rencker, F. Bach, W. Wang, and M. D. Plumbley, "Sparse recovery and dictionary learning from nonlinear compressive measurements," *IEEE Transactions on Signal Processing*, vol. 67, no. 21, pp. 5659–5670, 2019.
- [28] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.
- [29] S. Ravishanker and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 3088–3092.
- [30] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [31] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cospase analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.
- [32] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM J. Imaging Sciences*, vol. 1, pp. 228–247, Jan. 2008. DOI: 10.1137/07070156X.
- [33] S. Hawe, M. Kleinstueber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.
- [34] S. Ravishanker and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, 2012.
- [35] J. Dong, W. Wang, and W. Dai, "Analysis SimCO: A new algorithm for analysis dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 7193–7197.
- [36] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (SimCO) for dictionary update and learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6340–6353, 2012.
- [37] D. Pejic and M. Arsic, *Minimization and Maximization of Functions: Golden-Section Search in One Dimension*. May 2019, pp. 55–90, ISBN: 978-3-030-13802-8. DOI: 10.1007/978-3-030-13803-5_3.
- [38] S. Kitić, N. Bertin, and R. Gribonval, "Audio declipping by cospase hard thresholding," *2nd International Traveling Workshop on Interactions between Sparse models and Technology*, Aug. 2014. [Online]. Available: <https://hal.inria.fr/hal-00922497>.
- [39] J. N. Laska, P. T. Boufounos, and R. Baraniuk, "Finite Range Scalar Quantization for Compressive Sensing," Marseille, France, Tech. Rep., May 2009, Special session on sampling and quantization. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00452261>.
- [40] S. Kitić, L. Jacques, N. Madhu, M. P. Hopwood, A. Spriet, and C. De Vleeschouwer, "Consistent iterative hard thresholding for signal declipping," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 5939–5943.
- [41] K. Siedenburg, M. Kowalski, and M. Dörfler, "Audio declipping with social sparsity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 1577–1581.
- [42] B. Defraene, N. Mansour, S. De Hertogh, T. Van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2627–2637, 2013.
- [43] S. Foucart and T. Needham, "Sparse recovery from saturated measurements," *Information and Inference: A Journal of the IMA*, vol. 6, no. 2, pp. 196–212, 2017.
- [44] P. Smaragdis, "Dynamic range extension using interleaved gains," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 966–973, 2009.
- [45] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [46] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.
- [47] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, Springer, 2011, pp. 185–212.
- [48] S. B. Neal Parikh, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [49] J. Eckstein and D. P. Bertsekas, "On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1–3, pp. 293–318, 1992.
- [50] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2017, pp. 323–332.
- [51] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF: A new sparse representation for acoustic signals," Apr. 2009, pp. 3437–3440. DOI: 10.1109/ICASSP.2009.4960364.
- [52] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Projective dictionary pair learning for pattern classification," *Advances in Neural Information Processing Systems*, vol. 1, pp. 793–801, 2014.
- [53] M. Sandbichler and K. Schnass, "Online and stable learning of analysis operators," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 41–53, 2019.

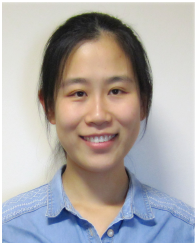


processing. His research interests include digital signal processing for image, radar and audio data.



Bin Li received the B.S. degree in communication engineering from Northwestern Polytechnical University, China, in 2002, and the M.E. degree in communication and information system from China Academy of Space Technology, in 2005. He has been working in China Academy of Space Technology since 2005, where he is currently a Senior Researcher. He was an Academic Visitor at the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, from 2019 to 2020, where he studied in the area of sparse representation for signal processing. His research interests include digital signal processing for image, radar and audio data.

Lucas Rencker graduated from Ecole Centrale Marseille, France, in 2015, and received the Ph.D. degree from the University of Surrey, UK, in 2020, where he worked in the Centre for Vision, Speech and Signal Processing (CVSSP), as part of the "Machine Sensing" Innovative Training Network (MacSeNet). His research interests include sparse coding, compressed sensing and dictionary learning, with application to inverse problems in signal processing.



Jing Dong received the B.E. degree in automation in 2010 and the M.E. degree in navigation guidance and control in 2013, both from Harbin Engineering University, Harbin, China. She received the Ph.D. degree in electronic engineering from University of Surrey, Guildford, U.K. in 2016. She is currently an Associate Professor at Nanjing Tech University. Her current research interests include dictionary learning, image processing and machine learning.



Yuhui Luo (M'02) graduated with Ph.D. degree in Signal Processing and Communications from Imperial College London, UK in 2002. From 2002-2004, she worked as postdoctoral researcher in Centre for DSP Research at King's College London, UK. In 2005, she joined the Core Technology Group of Samsung Research Institute (SERI) UK as Software Engineer. From 2006 -2015, she worked in the fields of financial futures trading and product management. Her positions include Quantitative Researcher at Winton Capital Management, Senior Researcher at China Merchant Futures, and Senior Quantitative Product Manager at Guosen Securities. From 2017, she has been working as Senior Scientist at Data Science Departments of National Physical Laboratory (NPL), UK. Her research interests are in uncertainty measurement, machine learning and signal processing related applications.



Mark Plumbley (S'88-M'90-SM'12-F'15) received the B.A.(Hons.) degree in electrical sciences and the Ph.D. degree in neural networks from University of Cambridge, Cambridge, U.K., in 1984 and 1991, respectively. Following his PhD, he became a Lecturer at King's College London, before moving to Queen Mary University of London in 2002, where he subsequently became Professor and Director of the Centre for Digital Music. He joined the University of Surrey in 2015 as Professor of Signal Processing, and in 2019 became Head of School of Computer

Science and Electronic Engineering. He is known for his work on analysis and processing of audio and music, using a wide range of signal processing techniques, including matrix factorization, sparse representations, and deep learning, and holds an EPSRC Fellowship on AI for Sound. He was a co-editor of a recent book on Computational Analysis of Sound Scenes and Events, and was Co-Chair of the recent DCASE 2018 Workshop on Detection and Classifications of Acoustic Scenes and Events. He is a Member of the IEEE Signal Processing Society Technical Committee on Audio and Acoustic Signal Processing, and a Fellow of the IET and IEEE.



Wenwu Wang (M'02-SM'11) received the B.Sc. degree in 1997, the M.E. degree in 2000, and the Ph.D. degree in 2002, all from the College of Automation, Harbin Engineering University, China.

He then worked in King's College London (2002-2003), Cardiff University (2004-2005), Tao Group Ltd. (now Antix Labs Ltd.) (2005-2006), Creative Labs (2006-2007), before joining University of Surrey in May 2007, where he is a Professor in Signal Processing and Machine Learning, and a Co-Director of the Machine Audition Lab within the Centre for Vision Speech and Signal Processing. His current research interests include signal processing, machine learning and perception, with a focus on audio/speech and multimodal (e.g. audio-visual) data. He has (co)-authored over 250 publications in these areas. He is a (co)-recipient of more than 10 awards, including the Judge's Award on DCASE 2020, Reproducible System Award on DCASE 2019 and DCASE 2020, and Best Student Paper Award on LVA/ICA 2018.

He has been a Senior Area Editor (2019-) and Associate Editor (2014-2018) for IEEE Transactions on Signal Processing. He is an Associate Editor (2020-) for IEEE/ACM Transactions on Audio Speech and Language Processing. He is a Specialty Editor-in-Chief of Frontiers in Signal Processing since 2020. He was a Publication Co-Chair for ICASSP 2019, Brighton, UK. He has been elected as a Member (2021-) of the IEEE Signal Processing Theory and Methods Technical Committee and a Member (2021-) of the IEEE Machine Learning for Signal Processing Technical Committee. He also serves as a Member (2019-) of the International Steering Committee of Latent Variable Analysis and Signal Separation.