



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna Archivio istituzionale della ricerca

Deep Neural Oracles for Short-Window Optimized Compressed Sensing of Biosignals

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Mangia M., Prono L., Marchioni A., Pareschi F., Rovatti R., Setti G. (2020). Deep Neural Oracles for Short-Window Optimized Compressed Sensing of Biosignals. IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, 14(3), 545-557 [10.1109/TBCAS.2020.2982824].

Availability:

This version is available at: <https://hdl.handle.net/11585/763624> since: 2020-06-30

Published:

DOI: <http://doi.org/10.1109/TBCAS.2020.2982824>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

Deep Neural Oracles for Short-window Optimized Compressed Sensing of Biosignals

Mauro Mangia, *Member, IEEE*, Luciano Prono, *Student Member, IEEE*, Alex Marchioni, *Student Member, IEEE*, Fabio Pareschi, *Senior Member, IEEE*, Riccardo Rovatti, *Fellow, IEEE*, and Gianluca Setti, *Fellow, IEEE*

Abstract—The recovery of sparse signals given their linear mapping on lower-dimensional spaces can be partitioned into a support estimation phase and a coefficient estimation phase. We propose to estimate the support with an oracle based on a deep neural network trained jointly with the linear mapping at the encoder. The divination of the oracle is then used to estimate the coefficients by pseudo-inversion. This architecture allows the definition of an encoding-decoding scheme with state-of-the-art recovery capabilities when applied to biological signals such as ECG and EEG, thus allowing extremely low-complex encoders. As an additional feature, oracle-based recovery is able to self-assess, by indicating with remarkable accuracy chunks of signals that may have been reconstructed with a non-satisfactory quality. This self-assessment capability is unique in the CS literature and paves the way for further improvements depending on the requirements of the specific application. As an example, our scheme is able to satisfyingly compress by a factor of 2.67 an ECG or EEG signal with a complexity equivalent to only 24 signed sums per processed sample.

Index Terms—Compressed sensing, Biosignal compression, Low-complexity compression, Deep neural networks

I. INTRODUCTION

COMPRESSED Sensing (CS) is a relatively new paradigm for the acquisition/sampling of signals that violates the intuition behind the theorem of Shannon [1]–[3]. In fact, CS theory states that, under surprisingly broad conditions it is possible to reconstruct certain signals or images using far fewer samples or measurements than they are used with traditional methods. To enable this, CS is based on two concepts: *sparsity*, which is related to the signals of interest, and *incoherence*, which relates to the methods of measurement/acquisition/sampling. Sparsity expresses the idea many natural signals have a very parsimonious representation when expressed in an appropriate *sparsity basis*. Incoherence expresses the idea that a reduced number of acquisitions of a waveform that have a sparse representation in an appropriate basis, which is made in a domain that is incoherent with it, allows to always capture the entire signal information.

M. Mangia, A. Marchioni and R. Rovatti are with the Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40136 Bologna, Italy, and also with the Advanced Research Center on Electronic Systems, University of Bologna, 40125 Bologna, Italy (e-mail: alex.marchino@unibo.it, mauro.mangia2@unibo.it, riccardo.rovatti@unibo.it).

L. Prono is with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy (e-mail: luciano.prono@polito.it).

F. Pareschi and G. Setti are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy, and also with the Advanced Research Center on Electronic Systems (ARCES), University of Bologna, 40125 Bologna, Italy (e-mail: fabio.pareschi@polito.it; gianluca.setti@polito.it).

Based on these concepts, it is has been possible to devise protocols for sampling/measurement [4], [5] which capture the information content, but require a number of measurements comparable to the number of non-zero coefficients in the expression of the signal of interest with respect to its appropriate sparsity base. Consequently, the most significant feature of these sampling procedures is that they allow a sensor to capture the information content of a signal without going through the acquisition of its entire profile, thus performing acquisition and compression at the same time. In other words, CS is a very simple and efficient procedure to sample sparse signals at a reduced rate, using much less resources with respect to standard sampling required for A/D conversion.

Of particular interest is that many signals of interest in biomedical applications enjoy the sparsity property and can therefore be efficiently acquired using CS, i.e. by using less energy, in less time and/or with less samples. For example, this has been demonstrated for Electrocardiographic (ECG), Electromyographic (EMG) [6] and Electroencephalography (EEG) [7] signals, which paved the way to the adoption of CS for efficient acquisition of biosignals in Body Area Networks nodes [8], [9]. This has been shown as well for waveforms acquired through magnetic resonance imaging (MRI) [10] where by using CS one can obtain the very important results to accelerate the overall MRI acquisition [11].

All these advantages in the acquisition phase are balanced by the increase in complexity necessary for the signal reconstruction with respect to the simple low-pass filtering needed in a standard D/A conversion. In fact, reconstruction in a CS frameworks boils down to solving the problem (which is also fundamental in a number of heterogeneous applications) of recovering an n -dimensional sparse signal x from a set of m measurements y that represent the output of the CS under-sampling encoding, i.e., with $m < n$. More specifically, one needs to find the sparsest n -dimensional vector x among the infinite solution of the hill-defined system $y = \mathcal{L}(x)$, where $\mathcal{L} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a linear dimensionality-reduction operator, which, regrettably, is an NP-hard problem. Yet, thanks to [12], the solution can be obtained by solving a minimization problem, called *Basis Pursuit* (BP)¹, using linear programming. In other words, the result in [12] is fundamental since it allows to obtain a solution for the BP problem in polynomial time, thus making the use of CS practical. Yet, the computational resources needed by the numerical

¹The problem is called *Basis Pursuit with DeNoising* (BPDN) if noise is also considered

algorithm solving BP may be so demanding to make its solution practically unfeasible in low-complexity nodes, like a typical BAN gateway. To cope with this, several dedicated BP/BPDN solvers have been proposed such as the Spectral Projected Gradient for L1 Minimization (SPGL1) [13], and the Generalized Approximate Message Passing (GAMP) [14]. Alternative solutions rely on the observation that the main issue in the computation of x is not finding a generic solution to $y = \mathcal{L}(x)$, but to find the sparse one. Starting from this, further computational cost reduction can be achieved by generating solutions which iteratively adjust their sparsity at each step. Different heuristics may be used to promote sparsity and give raise to different methods, such as the Orthogonal Matching Pursuit (OMP) [15] and the Compressive Sampling Matching Pursuit (CoSaMP) [16].

More recently it has been demonstrated that additional advantages in terms of a smaller computational complexity or improvement in the quality of the reconstructed signal can be obtained by adopting a (Deep) Neural Network (DNN) for reconstruction [17]–[24]. More specifically, in [21] authors have shown a probabilistic relation between CS and a stacked denoising autoencoder (SDA) implemented using a 3-layer simple structured neural network. Once properly trained, the SDA has been capable to directly recover a sparse image from its linear (or mildly non-linear) measurements, and has offered, in some cases, advantages in terms of quality of the reconstructed images with respect to the most common greedy reconstruction algorithms. A similar approach which employs fully-connected DNNs can be found in [22], where CS has been applied to videos and the proposed approach enables fast recovery of video frames at significantly improved reconstruction quality. In [23] authors have proposed a DNN, called ISTA-Net and inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [25], which has been designed with the aim of optimizing the solution of BP to reconstruct compressed images. Another deep learning framework (referred as BW-NQ-DNN) applied to CS acquisition/reconstruction of neural signals has been presented in [24], where three networks have been jointly optimized for performing a binary measurement matrix multiplication, a non-uniform quantization and reconstruction, respectively. Despite the advantage shown in terms of quality of reconstruction, this approach has the drawback in terms of complexity of the resulting system, and in particular of the DNN used for the non-uniform signal re-quantization.

The aim of this work is to propose an innovative use of DNNs in a CS-based acquisition/reconstruction framework. More specifically our network is not used, as it is the case for all the above mentioned Literature, to directly reconstruct the input signal, but only to provide a *divination* of the support of the input signal, i.e. of the positions of the non-null components of the original signal when it is expressed along the sparsity basis. Our approach not only improves reconstruction quality with respect to standard techniques, but also introduces a *self-assessment capability* that allows to estimate *on the fly* the quality of reconstruction. Furthermore, with our approach signals can be successfully reconstructed even when they have been sampled using CS referring to very short acquisition windows. This is important since it

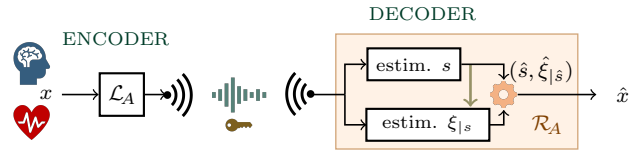


Fig. 1. General scheme of an encoder-decoder pair based on CS. In the decoder we distinguish the estimation of the signal support s and the estimation of the non-zero coefficients $\xi_{|s}$. Classical decoders perform both estimation simultaneously. Our approach first estimates s and then $\xi_{|s}$.

allows to further reduce the complexity of the acquisition stage and/or allow aggressive mixed-signal implementation of the acquisition stage.

To the best of our knowledge, this is the first work proposing to use a DNN for support identification, and one of the few proposing to use a DNN to improve reconstruction of signals sampled using CS which are not images.

The rest of the paper is organized as follows. Section II introduces some basic concepts of the CS. In Section III the choice of n is analyzed with pros and cons for the two considered classes of signals, ECGs and EEGs. Section IV recaps standard and oracle-based CS decoders while in Section V the adopted figures of merit are defined. Section VI introduces the DNN architecture as main building block of the proposed CS decoder described in Section VII. The same section includes performance analysis and comparisons with other standard CS decoders. The self-assessment capability is the topic of Section VIII while Section IX reports computational analysis for both encoder and decoder. Finally, we draw the conclusion.

II. COMPRESSED SENSING BASICS

Let us refer to the scheme in Figure 1 and assume to work by chopping input waveforms into subsequent windows, each of which is represented by a set of its samples $x = (x_0, \dots, x_{n-1})$ collected at Nyquist rate that we see as a vector $x \in \mathbb{R}^n$. CS hinges on the assumption that x is κ -sparse, i.e., in the simplest possible setting, that an orthonormal matrix S exists (whose columns are the vectors of the sparsity basis) such that when we express $x = S\xi$, then the vector $\xi = (\xi_0, \dots, \xi_{n-1})$ does not contain more than $\kappa < n$ non-zero entries.

The fact that x depends only on a number of scalars that is less than its sheer dimensionality hints at the possibility of compressing it. CS does this by applying a linear operator $\mathcal{L}_A : \mathbb{R}^n \mapsto \mathbb{R}^m$ depending on the acquisition (or encoding) matrix $A \in \mathbb{R}^{m \times n}$ with $m < n$ and defined in such a way that $x \in \mathbb{R}^n$ can be retrieved from $y = \mathcal{L}_A(x) \in \mathbb{R}^m$. The ratio n/m is the *compression ratio* and will be indicated by CR.

It can be intuitively accepted that the larger the κ , the larger the m is needed to guarantee that x can be retrieved from y and thus the lower the achievable CR. This relationship is asymptotically identified by CS theory as $m = \mathcal{O}(\kappa \log(n/\kappa))$ [2]. In finite and practical cases, one may often aim at using an m value proportional to κ though the most elementary, worst-case theoretical guarantees fail for $m < 2\kappa$. In fact, though $y = \mathcal{L}_A(\xi)$ has an infinite number of counterimages,

the first prerequisite for the recoverability is that when we add the κ -sparsity prior only one of them survives. Hence, given any two κ -sparse vectors ξ' and ξ'' it cannot be $y = \mathcal{L}_A(\xi')$ and $y = \mathcal{L}_A(\xi'')$, i.e., $\mathcal{L}_A(\xi' - \xi'')$ must be non-zero. Hence, $\xi' - \xi''$ cannot be in the kernel of \mathcal{L}_A . Since, in the worst-case, $\xi' - \xi''$ is 2κ -sparse, the only way of guaranteeing this is that \mathcal{L}_A when restricted to any 2κ -dimensional coordinate subspace of \mathbb{R}^n is a maximum rank operator. Clearly, if $\mathcal{L}_A : \mathbb{R}^n \mapsto \mathbb{R}^m$ with $m < 2\kappa$ this is not possible and, whenever the worst-case scenario is hit, the sparsity prior is no longer able to guarantee signal recovery. In practice, though worst case scenarios seldom appear, classical reconstruction algorithms fail before the limit $m = 2\kappa$ is reached.

Clearly, compression by \mathcal{L}_A must be coupled with a signal reconstruction stage² $\mathcal{R}_A : \mathbb{R}^m \mapsto \mathbb{R}^n$ such that ideally $x = \mathcal{R}_A(\mathcal{L}_A(x))$. In practice the chain of the encoding and decoding step is a lossy process and $\hat{x} = \mathcal{R}_A(\mathcal{L}_A(x))$ is only an approximation of x .

III. SIGNAL ENCODING AND PROS/CONS OF SHORT WINDOWS

The class of linear operators \mathcal{L}_A that can be effectively paired with a decoder \mathcal{R}_A is extremely large. Most notably, if A is an instance of a matrix whose entries are independent zero-average and unit-variance Gaussian random variables, then $\mathcal{L}_A(x) = Ax$ is known to work [1], [2], [26] with very high probability. Yet, if the matrix A^\pm is defined as $A_{j,k}^\pm = \text{sign}(A_{j,k})$, then $\mathcal{L}_A(x) = A^\pm x$ is also known to work with very high probability [27]. In the following, we will focus on $\mathcal{L}_A(x) = A^\pm x$ as this makes the computation of $\mathcal{L}_A(x)$ multiplierless and is thus the best option for very low resources implementations of the encoder stage.

Actually, the Literature shows that there is plenty of room for optimizing A [28]–[31], and suitably designed matrices are able to increase compression considerably with respect to naive random instances.

Clearly, this paves the way to applications in all those settings in which the computational complexity of compression must be kept at bay, e.g., in BANs for which reduced computation and compression before transmission are essential to fit within a tight resource budget.

It is worth stressing that, to best express its potential in reducing computational complexity at the encoder, CS should consider the shortest possible acquisition windows.

To understand why, consider the processing of N given samples. They may be partitioned into N/n contiguous and non-overlapping time windows, each with n samples. Operator \mathcal{L}_A can be applied to each window, entailing a number of operations $\mathcal{O}(n \cdot m)$. The total number of operations to process the N samples is $\mathcal{O}(n \cdot m \cdot N/n) = \mathcal{O}(n \cdot N/\text{CR})$.

Yet, CR is fixed to a sufficient level to reconstruct the original n -dimensional signal x from the m measurement y with a quality that is deemed acceptable. Hence, at given CR and N , the computational complexity is linearly increasing with n , i.e., with the length of individual time windows.

²Terms like *decoding* or *recovery* are also used to describe this stage.

Another aspect that has to be considered is the signal reconstruction latency. Even considering that $\mathcal{R}_A(y)$ is an instantaneous operation, the reconstructed signal is recovered with a delay of up to n time steps, since y is available with a delay of up to n time steps. Of course, the smaller the n , the lower the reconstruction latency.

Beyond this high-level reasons, short windows may benefit the implementation of the encoder also at a more physical level.

In purely digital realizations [32]–[34], the samples come from a conventional Analog-to-Digital converter and the encoder is implemented as a sequence of sums and subtractions depending on the entries of A^\pm . In this case, not only the computation time but also the memory needed to store A reduces when n (and m) gets smaller.

In mixed-mode realizations (i.e., in the design of Analog-to-Information converter based on CS) [8], [9], [35]–[37], $y = A^\pm x$ is computed component-wise as $y_j = \sum_{k=0}^{n-1} A_{j,k}^\pm x_k$, i.e., accumulating the signal samples in the analog domain. This implies an analog storage to hold the intermediate sum value. Yet, independently of the actual implementation and technology, the approach is doomed to suffer from leakage and disturbance [9], [38]. These phenomena degrade the stored value along time and their effect increases with the hold time and the number of sums. Hence, the lower the n , the shorter the time and the smaller number of operations needed to compute y_j , and therefore the smaller the degradation incurred before conversion into digital words occurs.

Regrettably, gaining all the advantages connected with the reduction of n is not straightforward. In fact, real world signals are such that, when n shrinks, the ratio κ/n tends to increase. As κ affects m , any reduction of n tends to impair the compression ratio.

To get a quantitative feeling of these trends, we show in Figure 2 the normalized sparsity κ/n for different values of n observed in the classes of ECG and EEG signals. Instances are obtained according to the synthetic generators described in the Appendix. Moreover, for both classes of bio-signals the considered sparsity basis S is a family of the orthogonal Wavelet functions [39]. In more detail, we select the Symmlet-6 family as sparsity basis for ECG signals [9], while our choice for the EEG case is the Daubechies-4 family [40].

The value of κ is computed as the lowest number of entries of the sparse representation that includes the 99.5% of the energy in the 99% of the ECG instances, and the 95% of the energy in the 99% of the EEG instances³. From the figure it is clear that the smaller the n , the larger the (normalized) sparsity, and therefore the lower the attainable CR that ensures a target reconstruction quality.

The above considerations reveal that there is a multi-faceted trade-off linking computational/implementation complexity, reconstruction quality and compression.

³For the class of EEG signals we refer to a synthetic signal that emulates event-related brain potentials, where readings in each lead contain information on the external stimulus as well as a part on other neurons activity. This is why we assume that 95% of the total energy is enough to identify the meaningful components of the signal.

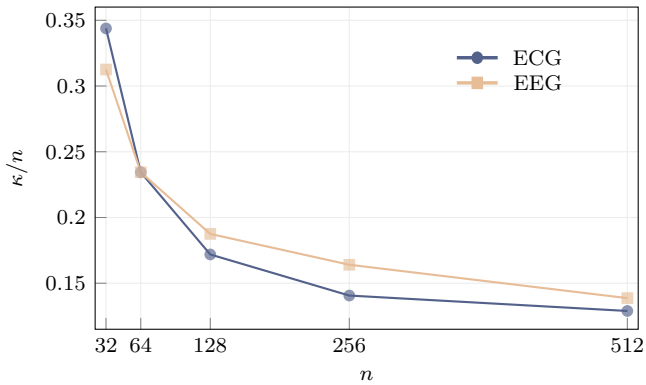


Fig. 2. The effect of reducing n on the normalized sparsity κ/n in the two examples of a synthetic ECG and a synthetic EEG signal.

The joint design of the encoder and of the oracle-based decoder we propose *tries to address such a trade-off in a better way with respect to conventional approaches.*

IV. SIGNAL RECOVERY AND SUPPORT ORACLES

To better formalize sparsity and its consequences, recall $x = S\xi$ and that not more than κ entries of ξ are non-null. The positions of the non-zero entries of ξ identify the so-called *support* $\text{supp } \xi$ that we will represent by means of the binary vector $s \in \{0, 1\}^n$ such that $s_j = 1$ if $\xi_j \neq 0$ and $s_j = 0$ otherwise. Binary, n -dimensional vectors can be used to *index* a generic n -dimensional vector v so that $v_{|s}$ is the subvector of v collecting only the entries v_j such that $s_j = 1$. We will use binary n -dimensional vectors also to *index* matrices M with n columns so that $M_{|s}$ is the submatrix of M that contains only the columns whose index j is such that $s_j = 1$.

With this notation, κ -sparsity is equivalent to say that x is efficiently represented by two pieces of information, namely the n -dimensional binary vector s and the real vector $\xi_{|s}$ whose dimensionality does not exceed κ .

Sparsity is fundamental in the decoding process going from y back to x . In fact, since $m < n$, the mapping $y = A^\pm S\xi$ from ξ to y is non-injective. Hence, any given measurement vector y corresponds to an infinite number of possible ξ . Yet, if A is properly designed, only one of the counterimages of y is κ -sparse and can be found by relatively simple algorithmic means.

As shown in Figure 1, a decoder recovers both s and ξ_s . Among the many methods proposed in the literature, the most classical approach is BPDN which recovers both pieces of information simultaneously by solving the optimization problem

$$\hat{\xi} = \arg \min_{\xi \in \mathbb{R}^n} \|\xi\|_1 \quad \text{s.t.} \quad \|y - A^\pm S\xi\|_2 \leq \tau \quad (1)$$

where $\hat{x} = S\hat{\xi}$ is the reconstructed signal, $\|v\|_p$ indicates the p -norm of the generic vector v , and $\tau \geq 0$ accounts for the possible presence of disturbances in the computation of y by relaxing the constraint $y = A^\pm S\xi$ that would hold in the noiseless case. The noiseless case itself corresponding to solve

the simpler BP problem can, of course, be tackled by setting $\tau = 0$.

Though implicitly performed, support identification is an important ingredient in BP and BPDN and is embedded in the 1-norm used in the objective function that, among all the possible ξ satisfying the constraint, selects the one with the largest number of zeros entries. This is so true that changing the 1-norm in the merit function would completely spoil reconstruction, while changing the 2-norm in the constraint usually still gives sensible results.

We here consider a different approach in which support identification is performed by an *oracle* looking at the vector y and divining s . Once s is known one may note that $y = A^\pm S\xi$ is a equivalent to $y = A^\pm S_{|s}\xi_{|s}$ to estimate $\xi_{|s}$.

The oracle is based on a Deep Neural Network (DNN) trained on signals with the same statistical features of the one to be acquired. Training involves also the matrix A , so that encoder and decoder are jointly optimized to improve support identification and thus to improve reconstruction performance.

V. PERFORMANCE INDEXES

The encoder-decoder chain may simultaneously perform more than one useful operation on the signal (see, e.g., [41]–[44] for its use as an encryption stage) of which compression is surely the most obvious as $m < n$. The compression performance of the encoder-decoder chain is easily assessed by the compression ratio n/m .

Yet, such a compression is in general lossy, and some degradation appears yielding $\hat{x} \neq x$. The closer \hat{x} to x , the better the encoder-decoder chain and this can be assessed by means of the Reconstruction Signal-to-Noise Ratio (RSNR) defined as

$$\text{RSNR} = \frac{\|x\|_2}{\|x - \hat{x}\|_2} \text{dB}$$

where for any scalar a , the a dB notation is equivalent to $20 \log_{10}(a)$. RSNR can be used to define two ensemble-level performance figures, computed starting from a set $x^{(t)}$ (for $t = 0, \dots, T-1$) of signal instances recovered as $\hat{x}^{(t)}$. The first is the Average RSNR (ARSNR)

$$\text{ARSNR} = \frac{1}{T} \sum_{t=0}^{T-1} \text{RSNR}^{(t)}$$

while the second is the Probability of Correct Reconstruction (PCR) estimated as

$$\text{PCR}(\text{RSNR}_{\min}) = \frac{1}{T} \# \left\{ t \mid \text{RSNR}^{(t)} \geq \text{RSNR}_{\min} \right\}$$

where $\#$ counts the number of elements in the set and RSNR_{\min} is the minimum RSNR level that is considered *sufficient* for a correct reconstruction.

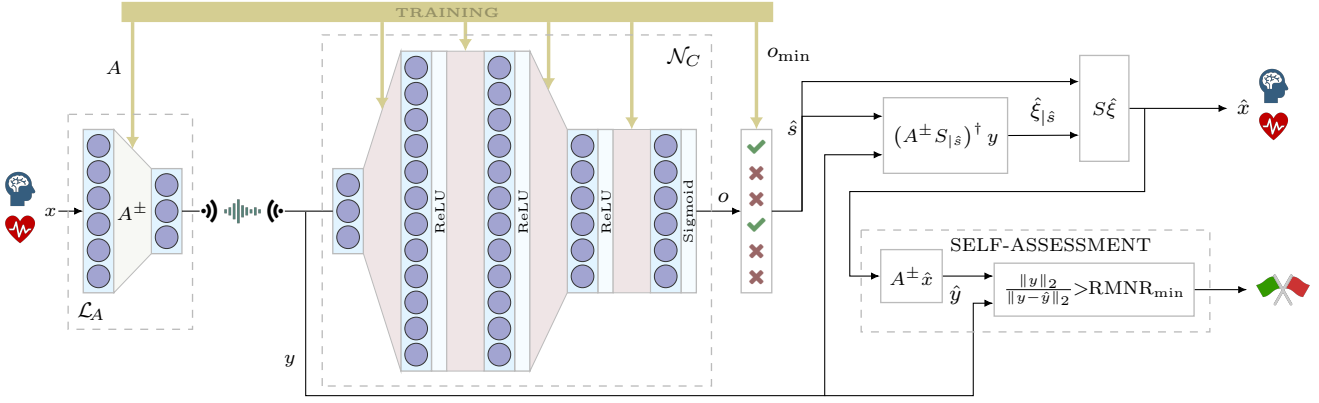


Fig. 3. Trained CS with support oracle block scheme including self-assessment capability. The fully connected DNN in the left is $\mathcal{N}_C \circ \mathcal{L}_A$ with an output layer $o \in [0, 1]^n$. Estimated support \hat{s} is such that $\hat{s}_j = 1$ if $o_j \geq o_{\min}$ and $\hat{s}_j = 0$ otherwise. The estimated support is employed in the signal reconstruction where the reconstructed signal \hat{x} is also the input of the self-assessment block.

VI. ORACLE STRUCTURE AND TRAINING

The core of the oracle is a Deep Neural Network (DNN) $\mathcal{N}_C : \mathbb{R}^m \mapsto [0, 1]^n$, defined by the connection parameters in C , with m inputs that correspond to the m entries of the measurement vector y and n outputs.

The neural network has 3 intermediate fully connected layers of cardinality $2n$, $2n$, and n , all with a ReLU activation function. The output layer is also fully connected with n units and sigmoidal activation function that map any scalar a into $(1 + e^{-a})^{-1}$.

Both the connection parameters C and the matrix A are initialized as instances of independent zero-average unit-variance Gaussian random variable and adjusted by training the compound system $\mathcal{N}_C \circ \mathcal{L}_A : \mathbb{R}^n \mapsto [0, 1]^n$.

The training set is made of a sequence of κ -sparse signals $x^{(t)} = S\xi^{(t)}$ (for $t = 0, \dots, T-1$) and of corresponding binary vectors $s^{(t)}$.

The true support of $\xi^{(t)}$ encoded in $s^{(t)}$ and the output $o^{(t)} = \mathcal{N}_C(\mathcal{L}_A(x^{(t)}))$ of the DNN are compared by means of a loss function, which is a total component-wise clipped cross-entropy between s and o defined by

$$X^{(t)} = - \sum_{j|s_j^{(t)}=1} L_\epsilon(o_j^{(t)}) - \sum_{j|s_j^{(t)}=0} L_\epsilon(1 - o_j^{(t)})$$

where $L_\epsilon(\cdot) = \min\{\log_2(1 - \epsilon), \max\{\log_2(\epsilon), \log_2(\cdot)\}\}$ for a small ϵ .

Though $\mathcal{L}_A(x) = A^\pm x$ in the forward pass, to prevent the sign function from interrupting error backpropagation, in the backward pass we assume $\nabla_A \mathcal{L}_A(x) = \nabla_A(Ax)$. With this, since $A_{j,k}^\pm = \text{sign}(A_{j,k})$ for every j and k , the training acts on the continuous-valued parameters whose sign is used in feedforward computation.

Using the methods specified in the Appendix, we generate a dataset composed by 8×10^5 signal instances for both the ECG and the EEG case. Each dataset is split in 80% for training (training set) and 20% for performance assessment (validation set).

All models proposed in this paper are implemented and trained by means of the TensorFlow framework [45] with the help of the high-level API provided by Keras [46].

Training is performed with stochastic gradient descent, where each gradient step is computed with a mini-batch comprising of 30 signal instances and where the initial learning rate value is 0.1.

To appreciate the complexity of the networks we propose, the one for $n = 64$ and with m ranging in $[16, 40]$ contains from 32128 to 36736 parameters and in our examples is trained for 500 epochs⁴. The network for $n = 128$ and with m ranging in $[24, 64]$ contains from 124672 to 140032 parameters and in our examples is trained for 1000 epochs.

VII. TRAINED CS WITH SUPPORT ORACLE

The trained oracle can be exploited in the definition of the decoder reported in Figure 3. We compute $o = \mathcal{N}_C(y)$ and, given a certain threshold $o_{\min} \in [0, 1]$ we estimate s with the binary vector $\hat{s} \in \{0, 1\}^n$ such that $\hat{s}_j = 1$ if $o_j \geq o_{\min}$ and $\hat{s}_j = 0$ otherwise. Starting from \hat{s} we finally estimate

$$\hat{\xi}_{|\hat{s}} = (A^\pm S_{|\hat{s}})^\dagger y \quad (2)$$

where \cdot^\dagger indicates Moore-Penrose pseudo-inversion that is needed since the number of ones in \hat{s} is in the order of $\kappa < m$ and the matrix $A^\pm S_{|\hat{s}}$ is a *tall* matrix with more rows than columns. The two estimations \hat{s} and $\hat{\xi}_{|\hat{s}}$ define the recovered signal \hat{x} .

Decoder operations depend on the value of o_{\min} that is set by a further training phase in which each vector in the training set is encoded and decoded for different values of o_{\min} . The o_{\min} yielding the highest ARSNR is selected. We name our approach Trained CS with Support Oracle (TCSSO) to summarize its main features.

We compare the performance of TCSSO with that of some well-known methods.

Since TCSSO simultaneously adapts encoder and decoder, we pair some classical signal recovery algorithms with an established technique for the optimization of the matrix A^\pm that is able to cope with the antipodality constraint on the entries.

⁴In each epoch the training algorithm walks through the entire training set

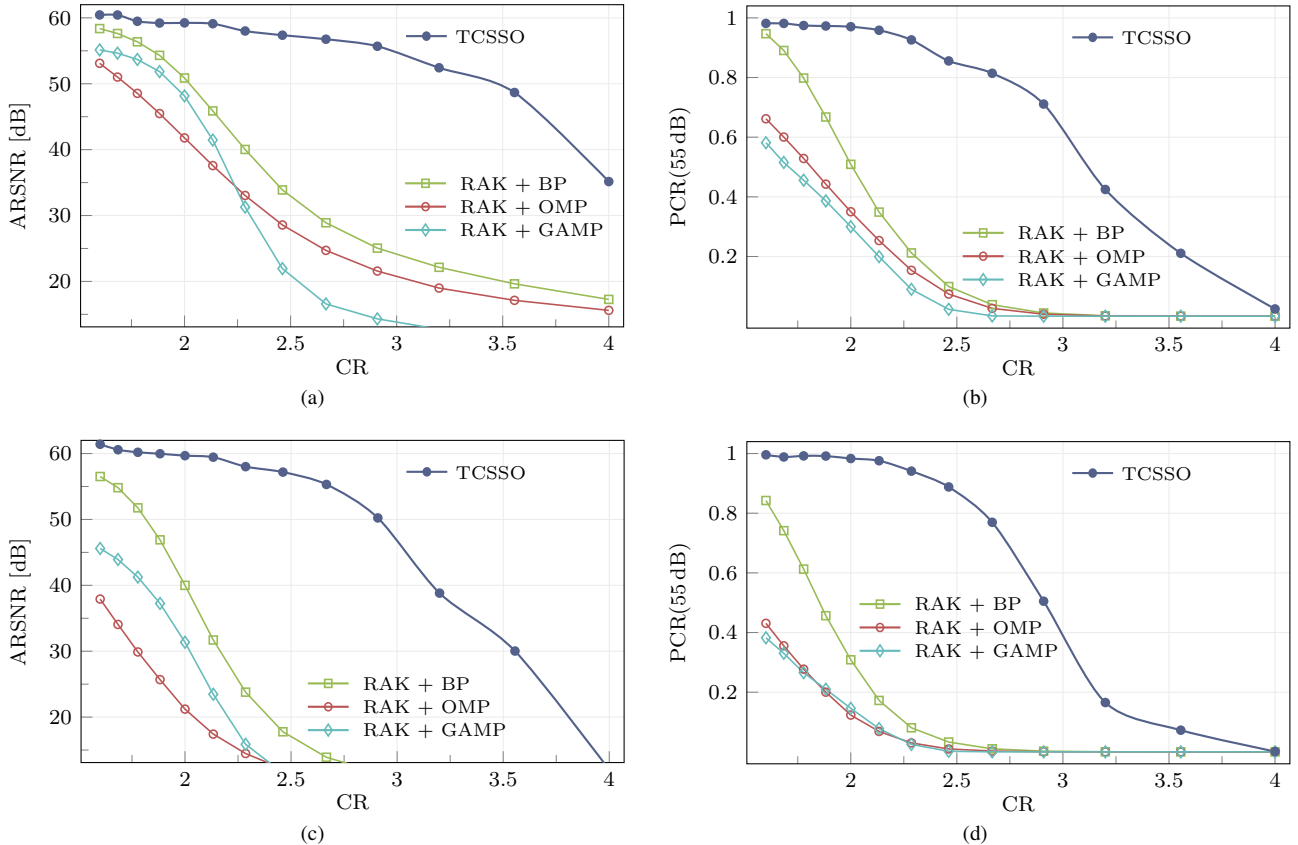


Fig. 4. Reconstruction performance for ECG (a)-(b) and EEG signals (c)-(d) in terms of both ARSNR (a)-(c) and PCR(55dB) (b)-(d) for the support oracle decoder with trained sensing matrices, TCSSO, against results for OMP, BP and GAMP with adapted sensing matrices (rakeness-based CS, RAK).

As decoders, we consider the lightweight OMP [15], BP that in all the tested cases has proven itself better than BPDN, and GAMP [14].

Matrix optimization is made by rakeness maximization [30], [31] that we have verified to yield much better results with respect to classical independent assignment of ± 1 to each of the entries of A^\pm .

We evaluate ARSNR and PCR by Montecarlo simulations using the samples of the validation set for both ECG and EEG cases with a superimposed noise equivalent to an Intrinsic Signal-to-Noise Ratio ISNR = 60 dB. The results are reported in Figure 4 for the $n = 64$ and $\kappa = 16$ case. In all plots, the number of measurements sweeps from $m = 40$ down to $m = 16$ thus focusing on compression ratios from CR = 1.6 up to CR = 4.

TCSSO clearly outperforms all other techniques and makes possible to work at compression ratios much larger than those commonly achievable though it still requires a very limited computational effort since $n = 64$.

Figure 5, show how the situation changes when n increases from 64 to 128. Performance is reported only in terms of ARSNR and increases with n as we are analyzing more data per chunk. The performance gap between TCSSO and the best of the classical methods also increases.

VIII. DECODER SELF-ASSESSMENT

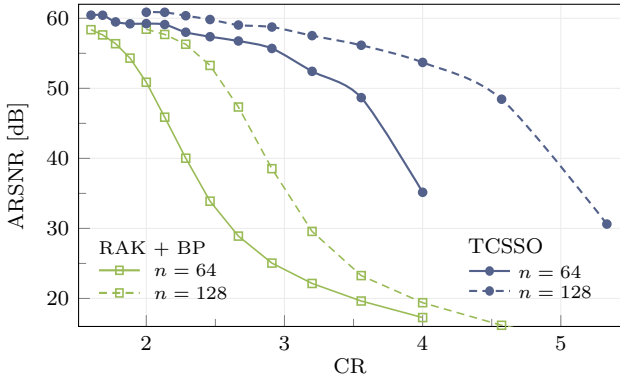
The TCSSO architecture described in the previous section can be extended by exploiting a property that stems from the fact that s is estimated separately from $\xi_{|s}$.

In fact, assume that no noise is present and that the size and content of A^\pm are such that $y = A^\pm S \xi$ is satisfied by one and only one κ -sparse ξ , i.e., that recovery of the true signal is theoretically possible. If the oracle is successful in divining the support, then $\hat{s} = s$ and $y = A^\pm S_{|\hat{s}} \xi_{|\hat{s}}$ implies that $y \in \text{span}(A^\pm S_{|\hat{s}})$, where for any matrix M , $\text{span}(M)$ is the subspace generated by the linear combination of its columns. This has a twofold consequence: i) (2) computes $\hat{\xi}_{|\hat{s}} = \xi_{|s}$, ii) if $\hat{\xi}$ is mapped back we have $A^\pm S \hat{\xi} = y$.

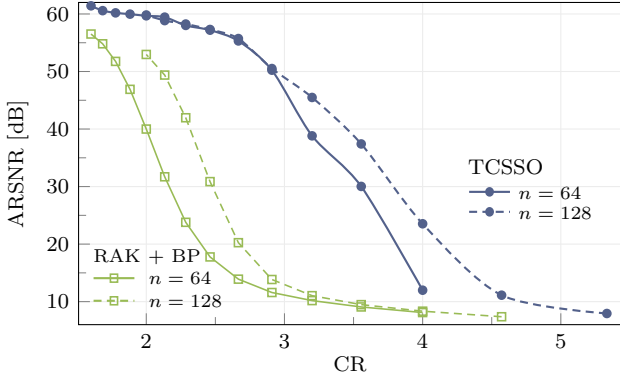
Yet, if the oracle fails, then $\hat{s} \neq s$ and since ξ is the unique κ -sparse solution of $y = A^\pm S \xi$ then $y \notin \text{span}(A^\pm S_{|\hat{s}})$. This has a twofold consequence: i) (2) computes $\hat{\xi}_{|\hat{s}} \neq \xi_{|s}$, ii) if $\hat{\xi}$ is mapped back we have $A^\pm S \hat{\xi} \neq y$.

Clearly, the decoder cannot check the correctness of $\hat{\xi}$ as the true ξ is unknown. Yet, it may map $\hat{\xi}$ back to measurement obtaining $\hat{y} = A^\pm S_{|\hat{s}} (A^\pm S_{|\hat{s}})^\dagger y = A^\pm \hat{x}$ that could be different from y . As a result, $\|y - \hat{y}\|_2$ is most naturally linked to the decoder failure and grants a useful self-assessment capability. In particular, one may monitor the quantity

$$\text{RMNR} = \frac{\|y\|_2}{\|y - \hat{y}\|_2}$$



(a)



(b)

Fig. 5. Performance in terms of ARSNR as function of the compression ratio for both TCSSO and the best observed traditional approach (RAK + BP). Results are for $n = 64$ as well as for $n = 128$ and for ECG (a) and EEG signals (b).

that is the Reconstruction Measurements-to-Noise Ratio, and declare that the oracle, and thus the TCSSO decoder, has succeeded when $\text{RMNR} \geq \text{RMNR}_{\min}$ for a certain threshold.

This situation can be exemplified in the small-dimensional case $n = 4$, $\kappa = 2$ and $m = 3$ with

$$A^\pm = \begin{pmatrix} +1 & +1 & +1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \end{pmatrix}$$

Since $\kappa = 2$ the instances of the original signal $\xi \in \mathbb{R}^4$ may have at most two non-null components and thus lay on the union of all the possible coordinate planes in \mathbb{R}^4 . We may indicate one of those planes as $c_{j,k}$ where j and k are the indexes of the non-null coordinates of its points. The matrix A^\pm is such that A^\pm maps each of those 6 coordinate planes into a plane in \mathbb{R}^3 that can be distinguished from the others. This is exemplified in Figure 6 on the left of which we draw the 6 planes $\iota_{j,k} \subset \mathbb{R}^3$ that are the images through $A^\pm S$ of the coordinate planes $c_{j,k} \subset \mathbb{R}^4$. Note that due to dimensionality reduction images are not pairwise orthogonal. Yet, recovery is theoretically possible as no two images $\iota_{j,k}$ and $\iota_{j',k'}$ are the same and thus a sufficiently clever algorithm can establish the support by looking at the measurement vector y .

Assume now that $s = (1, 1, 0, 0)$, i.e., that the true signal $\xi \in c_{0,1}$ is mapped by $A^\pm S$ into a measurement vector $y \in \iota_{0,1}$.

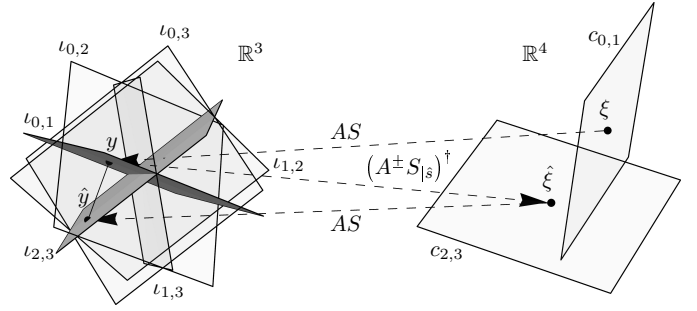


Fig. 6. The mechanism granting self-assessment capabilities to decoders based on a support oracle

Assume also that the oracle mistakes the support and estimates $\hat{s} = (0, 0, 1, 1)$, implying $\hat{\xi} \in c_{2,3}$.

By computing (2), the vector y is mapped back to $\hat{\xi}$ on that plane, that is therefore different from ξ . Though only approximately, the same holds in the noisy case and give an idea why the difference between y and \hat{y} assesses the correctness of the divined \hat{s} , i.e., the quality of the reconstruction \hat{x} .

As an example of the underlying mechanism, Figure 7 reports some Montecarlo evidence on the relationship between RMNR and RSNR for the ECG signals and in three different configurations. In Figure 7a no noise is present and $m = 32 = 2\kappa$; in Figure 7b $\text{ISNR} = 60$ dB and $m = 32 = 2\kappa$, whereas in Figure 7c no noise is present, but $m = 24 < 2\kappa$.

The two dimensional plots show an estimation of the joint-probability, conditioned to the *positive* events, i.e., the support has been correctly identified ($\hat{s}_j \geq s_j$ for all $j = 0, \dots, n-1$, orange points) or to the *negative* events, i.e., at least one entry in the support is neglected ($\hat{s}_j < s_j$ for at least one $j = 0, \dots, n-1$, blue points). Darker colors stand for higher densities.

The one dimensional plots at the bottom of the Figure report the error probabilities of a self-assessment procedure that calls for a positive event whenever $\text{RMNR} \geq \text{RMNR}_{\min}$ and for a negative event otherwise. As the threshold RMNR_{\min} increases, the probability of a false positive decreases since only very high RMNR reconstructions are declared correct. On the contrary, the probability of a false negative increases since for larger RMNR_{\min} even good reconstructions can be declared incorrect.

The ideal conditions in Figure 7a result in perfect self-assessment capabilities. When noise is added as in Figure 7b, positive and negative cases get mixed but remain identifiable by looking at RMNR.

Though no noise is present in Figure 7c, the fact that $m < 2\kappa$ makes the number of measurements insufficient, in general, for signal reconstruction, as there is no guarantee that only one κ -sparse signal ξ correspond to the given y through $A^\pm S$.

Hence, more than one support corresponding to the measurement exist. In these conditions, it may happen that the oracle divines a support that includes the true one (more than κ outputs of the network are larger than σ_{\min}) as well as components of other possible supports. In this case the oracle is not missing the support (orange point in the lower-right cluster in the scatter plot of Figure 7c) but pseudo-inversion

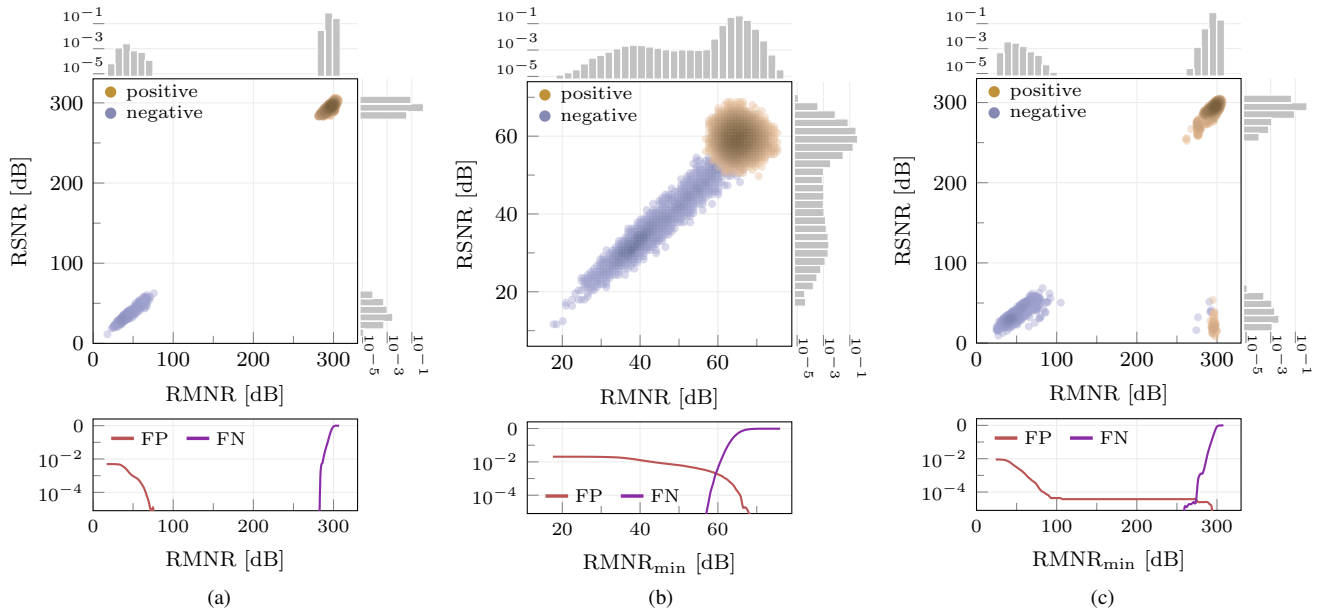


Fig. 7. The relationship between RSNR and RMNR for ECG signals. In all cases $n = 64$ and $\kappa = 16$. Orange dots correspond to cases in which \hat{s} includes all the components of s , while blue dots correspond to \hat{s} failing to identify some components in s . Above and to the right of the scatter plots are logarithmic histograms estimating the probability density of RMNR and RSNR. In (a) $m = 32 = 2\kappa$ and $\text{ISNR} = \infty$. In (b) $m = 32 = 2\kappa$ and $\text{ISNR} = 60$ dB. In (c), $m = 24 < 2\kappa$ and $\text{ISNR} = \infty$.

spreads the reconstruction over all the available components thus failing to reconstruct the signal. It may also happen that the oracle divines a support different from the true one. In this case the oracle is wrong (blue points in the lower-right cluster in the scatter plot of Figure 7c) and pseudo inversion identifies a sparse signal that is not the true one. Both cases give rise to points for which RMNR is very high but the RSNR is very low and no matter how high the RMNR_{\min} , the probability of a false positive is not vanishing.

Luckily enough, the above cases are the ones breaking *worst-case* guarantees and happen quite rarely: in our 1.6×10^5 validation set, for $n = 32$, $\kappa = 16$ and $m = 24$, the oracle divines a support in excess of the true one only 109 times, and a support different from the true one only 6 times. The statistics commonly used to assess performance remain substantially unaltered by these failures that are undetectable by looking at the RMNR.

In general, the value of RMNR_{\min} can be decided once that o_{\min} is set, by a further pass over the training set. This allows to estimate false positive and false negative curves as in Figure 7 and use them as criteria.

In the following we will set RMNR_{\min} as the largest value for which false negative probability is negligible.

Whenever a failure is detected, the decoder may take different actions whose effectiveness depend on the final applications.

The exploration of all the possibilities of the resulting two-level decoder is out of the scope of this paper but it can be easily recognized that quite a few options are available, such that:

- i) raise a warning and mark the current window as potentially incorrect;

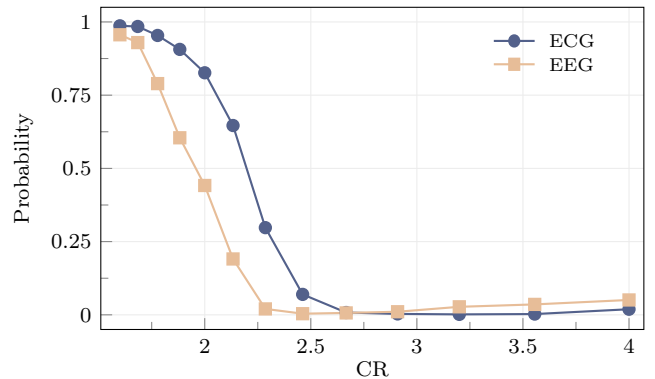


Fig. 8. Probability for GAMP decoder to reconstruct both ECG and EEG signals with RSNR higher than TCSSO in case of TCSSO failure. $\text{Prob}(\text{RSNR}_{\text{GAMP}} > \text{RSNR}_{\text{TCSSO}} \mid \text{RMNR} < \text{RMNR}_{\min})$.

- ii) feed the warning back to the encoder and require further information to correct the reconstruction (thus lowering the CR for this instance);
- iii) trigger another decoder on the same measurement vector hoping that this will improve reconstruction;
- iv) any combination of the above.

As a partial, non-optimized example, whose only aim is to show that some information can still be extracted from the measurements, when first-attempt TCSSO decoder fails, we trigger GAMP⁵ as a second-wind decoder.

Figure 8 plots the probability that GAMP yields an RSNR larger than what is given by TCSSO when applied to the

⁵GAMP has achieved better results compared to the other classical reconstruction algorithms in this setting, i.e., when the sensing matrix is not designed according to the rakeness-based CS.

instances that the latter marks as incorrectly recovered as $\text{RMNR} < \text{RMNR}_{\min}$, as a function of CR for the $n = 64$, $\kappa = 16$ case. A second-wind decoding is useful when such a probability is larger than 50%, i.e., approximately for $\text{CR} \leq 2$.

IX. COMPUTATIONAL REQUIREMENTS

As noted previously, CS-based lossy compression methods result in a multi-faceted trade-off between compression ratio, reconstruction quality, and computational complexity. In this section we give further detail on the last aspect, distinguishing what is required at the encoder (that we want to minimize) and at the decoder (that we want to be not worse than the needs of classical recovery methods). In all cases we refer to the computational burden *per processed sample*, i.e., we divide the number of operations by the number of samples n contained in the processed window.

A. Encoder

The complexity of the encoder is briefly introduced in Section III as one of the leading design criteria. The number of signed accumulations (AC) is $nm = n^2\text{CR}^{-1}$ thus yielding $n\text{CR}^{-1}$ AC/sample. Further to time-complexity, memory footprint is dominated by the storage of the matrix A^\pm , and requires a number of entries equal to $nm = n^2\text{CR}^{-1}$. In principle, matrix entries are bits. Yet, microcontroller-based implementations may favor 1-byte-per-entry or even 4-bytes-per-entry solutions. In fact, in some architectures the alignment of entries at word boundaries ensures better performance both in terms of speed and energy (see, e.g., [47]), this is why we express the memory footprint as the number of entries in A^\pm .

From the blue curves in Figure 5, one gets that an higher n results in better reconstruction performance for the same CR, and thus there is a trade-off between encoder complexity and window length.

Table I reports the comparison between the increase in terms of quality and the increase in terms of complexity and memory footprint for ECG signals when $n = 64$ and $n = 128$, and where CR ranges from 2 to 4. At high CR levels, increase in terms of quality (e.g., with $\text{CR} = 4$, +18.0 dB for ECG and +11.5 dB for EEG) may be worth the $\times 2$ in terms of computational effort and the $\times 4$ in terms of memory footprint. Yet, at lower values of CR, the increase in resource needs is not justified by the limited increase in performance: for $\text{CR} = 2$, resource requirement increase as before but one only gains +1.7 dB in the ECG case and +0.1 dB in the EEG case.

B. Decoder

In CS-based schemes, decoding is computationally much more intensive than encoding. We may evaluate the complexity of the TCSSO decoder by counting the number of Multiply-and-Accumulate (MAC) operations needed to compute \hat{x} , disregarding the training phase.

The oracle \mathcal{N}_C is composed by an input layer with m nodes, 3 fully connected hidden layers with $2n$, $2n$ and n nodes, and a final output fully connected layer with n nodes. The layer-by-layer number of MACs required for

TABLE I
PERFORMANCE IMPROVEMENT IN TERMS OF ARSNR, COMPUTATIONAL OVERHEAD IN TERMS OF AC/sample AND INCREASE IN MEMORY FOOTPRINT FOR THE SENSING MATRIX (# ENTRIES OF A^\pm) WITH n GOING FROM 64 TO 128. RESULTS ARE FOR TCSSO IN ECG AND EEG CASES WITH COMPRESSION RATIO (CR) RANGING FROM 2 TO 4.

CR	ARSNR [dB]				AC/sample		# $A_{j,k}^\pm$	
	ECG		EEG		64	128	64	128
	n	64	128	64				
4.00	35.2	+18.0	12.0	+11.5	16	$\times 2$	1.0 Ki	$\times 4$
3.20	52.4	+5.1	38.8	+6.7	20	$\times 2$	1.3 Ki	$\times 4$
2.67	56.8	+2.2	55.3	+0.4	24	$\times 2$	1.5 Ki	$\times 4$
2.29	58.0	+2.4	58.0	+0.2	28	$\times 2$	1.8 Ki	$\times 4$
2.00	59.2	+1.7	59.7	+0.1	32	$\times 2$	2.0 Ki	$\times 4$

the forward pass is $2nm$, $4n^2$, $2n^2$ and n^2 for a total of $(2m + 7n)n = (2/\text{CR} + 7)n^2$ MACs for each window thus yielding $(2/\text{CR} + 7)n$ MAC/sample.

After support estimation, calculations are needed to compute \hat{x} . In particular, one needs to compute the Moore-Penrose pseudoinverse of $B = A^\pm S_{|\mathcal{s}}^\pm$, i.e., of a matrix with m rows and a number c of columns $\kappa \leq c \leq n$, with $c \simeq \kappa$ being the most frequent case.

Though pseudo-inversion has optimized implementations, its typical complexity is equivalent to the computation of its analytical formula that, in our case, is $B^\top (BB^\top)^{-1}$. Considering that BB^\top is an $m \times m$ matrix whose inversion entails $2m^3$ MACs we arrive at estimating a total of $m(2m^2 + m\kappa + m + \kappa)$ for the typical $c = \kappa$ case. The complexity is $(2n\text{CR}^{-2} + n\kappa/n\text{CR}^{-1} + \kappa/n + \text{CR}^{-1})n\text{CR}^{-1}$ MAC/sample.

We may compare the complexity of TCSSO decoding with that of OMP, that is known to be one of the simplest and most light-weighted approaches. We consider the standard implementation of OMP as described in [48]. OMP begins with a number of iterations each of which entails some linear algebra and sorting whose aim is to add a component to the support. By focusing on matrix products and their cost in terms of number of MAC, the j -th step has a complexity equal to $nm + 2m(j - 1) + 2m + jm$. In the most economical implementation, it is possible to fix the number of iterations to κ and spare the checks needed to terminate the loop. This yields a total of at least $3\kappa m/2 + 3\kappa^2 m/2 + \kappa nm$ MACs. After that, OMP computes the pseudo-inverse of a matrix of the same size as the $B = A^\pm S_{|\mathcal{s}}^\pm$ in TCSSO.

The total complexity of the iterative part is therefore given by $(3/2 + 3n\kappa/n/2 + n)n\kappa/n\text{CR}^{-1}$ MAC/sample and must be compared with the computational effort required by the oracle that is $(2/\text{CR} + 7)n$ MAC/sample.

Though the different contributions to the computational complexities computed above have quite dissimilar asymptotic behaviors, their magnitude in the small- n cases can be appreciated only by numerical evaluation. As an example, for $n = 64$, $\kappa = 16$ and $\text{CR} = 2$ (one of our ECG cases) the first part of OMP entails some 716 MAC/sample, the oracle in TCSSO requires some 512 MAC/sample, while the common

pseudo-inversion amounts to 1304 MAC/sample. As a further, somehow opposite, example, for $n = 128$, $\kappa = 26$ and $CR = 4$ (one of our EEG cases) the first part of OMP entails some 1095 MAC/sample, the oracle in TCSSO requires some 960 MAC/sample, while the common pseudo-inversion amounts to 735 MAC/sample.

In both cases the complexity of TCSSO and that of OMP are analogous thus showing that, though TCSSO allows to implement extremely lightweight encoders, the decoder does not have to compensate by increasing its computational requirements with respect to conventional decoders.

X. CONCLUSION

We propose a CS decoder that, starting from the compressed measurements, first guesses which components are non-zero in the sparse signal to recover, and then computes their magnitudes.

Support guessing is provided by a suitable DNN-based oracle that reveals extremely accurate, especially when trained together with the encoding matrix.

The resulting decoder largely outperforms classical approaches, even when they are paired with one of the most effective adaptation policies for the encoding matrix. This allows to apply CS to signal windows containing a limited number of samples. The adoption of short windows is extremely beneficial along many directions, one of the most remarkable being the computational complexity of the encoder. Yet, short windows are usually out of the reach of classical CS mechanisms as the sparsity assumption on which they hinge tends to fail when the dimensionality of the waveform to compress decreases. Hence, our proposal allows the implementation of extremely low complexity encoders that still feature remarkable compression capabilities.

Furthermore, the separation between support guessing and magnitude calculation allows our decoder to detect cases in which the reconstruction may be affected by significant errors, thus paving the way, for example, to additional processing that further increases the reconstruction performance.

We demonstrated the effectiveness of this novel approach addressing realistic ECG and EEG signals for which compression ratios in the excess of 2 can be reached with a computational burden not exceeding 32 signed sums per sample.

APPENDIX

GENERATION OF ECG AND EEG DATASETS

Due to the large number of signal instances needed, in general, to train a neural network, both in the ECG and in the EEG case we used a MATLAB code to generate synthetic instances of the two classes of signals.

As mentioned in Section III, ECGs exhibit sparsity with respect to the orthonormal set of vectors representing the Symmlet-6 wavelet family transformation. Here, κ is set on 16 for $n = 64$ and 24 for $n = 128$. For the EEG signals the sparse vectors ξ are with respect the basis representing the Daubechies-4 wavelet transformation where $k = \{16, 26\}$ matches $n = \{64, 128\}$.

A. ECG

The synthetic generator⁶ of ECGs is thoroughly discussed in [49]. Signals are generated as noiseless waveforms. The noisy cases are obtained by superimposing additive white Gaussian noise whose power is such that the intrinsic SNR, ISNR, is 60 dB.

The setup is the same detailed in [30]. The heart-beat rate is randomly set using an uniform distribution between 60 beat/min and 100 beat/min. We generate chunks of 2 s with a 256 sample/s sampling frequency, that are split into windows of n subsequent samples. For both $n = 64$ and $n = 128$ cases we generate 8×10^5 input vectors x such that the corresponding total number of signal chunks are 10^5 and 2×10^5 . These input vectors are randomly split in training set and test set where the latter contains the 20% of the total amount of vectors x .

B. EEG

The detailed description of the code to generate the synthetic EEG signal⁷ can be found in [50]. The generator emulates event-related brain potentials, modeling an evoked potential as the series of a positive and a negative peak occurring at a fixed time relative to the event. The peaks are added to uncorrelated background noise, whose power is set to a level such that the resulting signal is very similar to an EEG signal measured by a real scalp electrode. Though the software is able to generate all channels in a multi-electrode EEG according to the standard 10-20 system, we focus on the ‘‘Fz’’ electrode, since it is in proximity (but not exactly on the top) of the simulated source of the stimulus. The sampling rate is set to 1024 sample/s with a stimulus frequency of 1 Hz.

We generate tracks corresponding to 50 different patients by starting from the parameters used in [50] and adding a random uniformly distributed offset to each of them. The ranges of the offsets for the positive peaks are ± 16 samples for the position of the peak, ± 0.05 Hz for the peak frequency and ± 1 for the peak amplitude. Ranges for the random offsets for negative peaks are: ± 26 samples for the position of the peak, ± 1 Hz for the peak frequency and ± 4 for the peak amplitude.

Signal length for each patient is such that the total number of n -samples windows is 8×10^5 . After that, 20% of signal instances for each patient are randomly select to contribute at the test set while remaining 80% is for the training phase.

REFERENCES

- [1] D. L. Donoho, ‘‘Compressed sensing,’’ *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [2] E. J. Candes, J. Romberg, and T. Tao, ‘‘Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,’’ *IEEE Trans. on Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.
- [3] R. G. Baraniuk, E. Candes, R. Nowak, and M. Vetterli (Eds.), ‘‘Special issue on compressive sampling,’’ *IEEE Signal Processing Magazine*, vol. 25, no. 2, March 2008.

⁶The MATLAB code is freely available for download from the Physionet website at <http://physionet.org/content/ecgsyn/>

⁷The MATLAB code is freely available for download from the Medical Research Council Brain Network Dynamics Unit at the University of Oxford website at <http://data.mrc.ox.ac.uk/data-set/simulated-ecg-data-generator>

- [4] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma (Eds.), "Special issue on applications of sparse representation and compressive sensing," *Proceedings of the IEEE*, vol. 98, no. 6, June 2010.
- [5] D. Allstot, R. Rovatti, and G. Setti (Eds.), "Special issue on circuits, systems and algorithms for compressed sensing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, Sep. 2012.
- [6] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ecg and emg wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, April 2012.
- [7] Z. Zhang, T. P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ecg via block sparse bayesian learning," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, Feb 2013.
- [8] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed sensing analog front-end for bio-sensor applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, Feb 2014.
- [9] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, and G. Setti, "Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 149–162, Feb. 2016.
- [10] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 72–82, 2008.
- [11] O. Jaspán, R. Fleysler, and M. L. Lipton, "Compressed sensing MRI: a review of the clinical literature," *British Journal of Radiology*, vol. 88, no. 1056, 2015.
- [12] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [13] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," Jun. 2007, <http://www.cs.ubc.ca/labs/scl/spgl1>.
- [14] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *2011 IEEE International Symposium on Information Theory Proceedings*, July 2011, pp. 2168–2172.
- [15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [16] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [17] A. Mirrashid and A. A. Beheshti, "Compressed remote sensing by using deep learning," in *2018 9th International Symposium on Telecommunications (IST)*, Dec 2018, pp. 549–552.
- [18] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 449–458.
- [19] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 2272–2276.
- [20] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 877–882.
- [21] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2015, pp. 1336–1343.
- [22] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digital Signal Processing*, vol. 72, pp. 9 – 18, 2018.
- [23] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1828–1837.
- [24] B. Sun, H. Feng, K. Chen, and X. Zhu, "A deep learning framework of quantized compressed sensing for wireless neural recording," *IEEE Access*, vol. 4, pp. 5169–5178, 2016.
- [25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [26] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, *Adapted Compressed Sensing for Effective Hardware Implementations: A Design Flow for Signal-Level Optimization of Compressed Sensing Stages*. Springer International Publishing, 2018.
- [27] J. Haboba, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "A pragmatic look at some compressive sensing architectures with saturation and quantization," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 443–459, Sep. 2012.
- [28] M. Elad, "Optimized projections for compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5695–5702, 2007.
- [29] J. Xu, Y. Pi, and Z. Cao, "Optimized projection matrix for compressive sensing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 560349, 2010.
- [30] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1001–1014, May 2012.
- [31] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, "Rakeness-based design of low-complexity compressed sensing," *IEEE Trans. on Circuits and Systems I: Reg. Papers*, vol. 64, no. 5, pp. 1201–1213, 2017.
- [32] J. Zhang, Y. Suo, S. Mitra, S. P. Chin, S. Hsiao, R. F. Yazicioglu, T. D. Tran, and R. Etienne-Cummings, "An efficient and compact compressed sensing microsystem for implantable neural recordings," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 485–496, Aug 2014.
- [33] D. Bellasi, M. Crescentini, D. Cristaudo, A. Romani, M. Tartagni, and L. Benini, "A broadband multi-mode compressive sensing current sensor soc in 0.16 μ m cmos," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 105–118, Jan 2019.
- [34] D. E. Bellasi, R. Rovatti, L. Benini, and G. Setti, "A low-power architecture for punctured compressed sensing and estimation in wireless sensor-nodes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 5, pp. 1296–1305, May 2015.
- [35] J. Yoo, S. Becker, M. Loh, M. Monge, E. Candès, and A. Emami-Neyestanak, "A 100mhz-2ghz 12.5x sub-nyquist rate receiver in 90nm cmos," in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, Jun. 2012, pp. 31–34.
- [36] X. Chen, E. A. Sobhy, Z. Yu, S. Hoyos, J. Silva-Martinez, S. Palermo, and B. M. Sadler, "A sub-nyquist rate compressive sensing data acquisition front-end," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 542–551, Sep. 2012.
- [37] M. Shoaran, M. H. Kamal, C. Pollo, P. Vanderghenst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, Dec 2014.
- [38] C. Paolino, F. Pareschi, M. Mangia, R. Rovatti, and G. Setti, "A Practical Architecture for SAR-based ADCs with Embedded Compressed Sensing Capabilities," in *2019 15th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, July 2019.
- [39] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Access Online via Elsevier, 2008.
- [40] N. Bertoni, B. Senevirathna, F. Pareschi, M. Mangia, R. Rovatti, P. Abshire, J. Z. Simon, and G. Setti, "Low-power eeg monitor based on compressed sensing with compressed domain noise rejection," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 522–525.
- [41] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-complexity multiclass encryption by compressed sensing," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2183–2195, May 2015.
- [42] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "On known-plaintext attacks to a compressed sensing-based encryption: A quantitative analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2182–2195, Oct 2015.
- [43] T. Bianchi, V. Bioglio, and E. Magli, "Analysis of one-time random projections for privacy preserving compressed sensing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 313–327, Feb 2016.
- [44] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.
- [45] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.
- [46] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.
- [47] A. Marchioni, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Sparse sensing matrix based compressed sensing in low-power ECG sensor nodes," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct 2017, pp. 1–4.

- [48] F. Pareschi, M. Mangia, D. Bortolotti, A. Bartolini, L. Benini, R. Rovatti, and G. Setti, "Energy analysis of decoders for rakes-based compressed sensing of ecg signals," *IEEE Transactions on Biomedical Circuits and Systems*, vol. PP, no. 99, pp. 1–12, 2017.
- [49] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Trans. on Biom. Eng.*, vol. 50, no. 3, pp. 289–294, Mar. 2003.
- [50] N. Yeung, R. Bogacz, C. B. Holroyd, and J. D. Cohen, "Detection of synchronized oscillations in the electroencephalogram: An evaluation of methods," *Psychophysiology*, vol. 41, no. 6, pp. 822–832, 2004.