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

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 always to capture the entire signal information.

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 fewer resources compared 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 fewer 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 compared 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. However, 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

Manuscript received Mmmm dd, yyyy; revised Mmmm dd, yyyy.

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).

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

in polynomial time, thus making the use of CS practical. Yet, the computational resources needed by the numerical 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 adjusts 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].

Further to those methods, schemes have been proposed in which the recovery algorithm is adapted to the class of signals to acquire (see, e.g., [17]–[19] where the decoder stages are tuned on the reconstruction of ECGs). These schemes exploit statistical priors on the signal to favor reconstructions close to what is typical in the class of acquired signals.

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 [20]–[27]. More specifically, in [24], authors have shown a probabilistic relation between CS and a stacked denoising autoencoder (SDA) implemented as 3-layer neural network. Once adequately trained, the SDA can 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 compared to the most common greedy reconstruction algorithms. A similar approach that employs fully-connected DNNs can be found in [25], where CS has been applied to videos, and the proposed approach enables fast recovery of video frames at a significantly improved reconstruction quality. In [26], authors have proposed a DNN called ISTA-Net and inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [28], which has been designed to optimize the solution of BP to reconstruct compressed images. Another deep learning model (BW-NQ-DNN) applied to CS acquisition/reconstruction of neural recording has been presented in [27]. Here, three networks have been jointly optimized to perform 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 a few drawbacks: *i*) it requires a pre-processing stage detecting signal peaks, which adds complexity to the encoder and specializes it to spiky signals; *ii*) it quantizes CS measurements after a programmable non-linearity, which adds further complexity.

This work proposes an innovative use of DNNs in a CS-based acquisition/reconstruction framework. Unlike all the cases mentioned above that use DNNs to reconstruct the input signal directly, our model only provides a *divination* of the support of the input signal, i.e., the positions of the non-null components of the signal expressed along the sparsity basis. Our approach not only improves reconstruction quality compared to standard techniques but also introduces a *self-assessment capability* that allows estimating *on the fly* the quality of reconstruction. Furthermore, with our method, signals can be successfully reconstructed even when they refer to very short acquisition windows, a crucial feature that a further reduction of the complexity and a mixed-signal implementation of the acquisition stage.

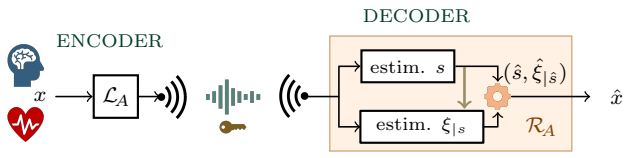


Fig. 1. General scheme of an encoder-decoder pair based on CS. In the decoder, we distinguish the estimation of the signal support s, and the estimation of the non-zero coefficients $\xi_{|s}$. Classical decoders perform both estimations simultaneously. Our approach first estimates s and then $\xi_{|s}$.

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 the 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 and introduces the proposed DNN-based oracle. The latter is the main building block of the proposed CS decoder, described in Section V along with performance analysis and comparisons with other CS frameworks. The self-assessment capability is the topic of Section VI, while Section VII reports computational analysis for both encoder and decoder. Finally, we draw the conclusion.

II. Compressed Sensing basics

Let us refer to the scheme in Fig. 1 and assume to work by chopping input waveforms into subsequent windows, each of which is represented by a set of its samples $x = (x_0, \ldots, x_{n-1})$ 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 $\xi = (\xi_0, \ldots, \xi_{n-1})$ does not contain more than $\kappa < n$ non-zero entries.

The fact that x depends only on a number of scalars that are 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 $m = \mathcal{O}(\kappa \log(n/k))$ [2]. In finite and practical cases, one may often aim at using a m value proportional to κ. Nevertheless, the worst-case theoretical guarantees fail for $m < 2\kappa$. In fact, despite the infinite number of counterimages of $y = \mathcal{L}_A(\xi)$, 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' an 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 stage[2] $\mathcal{R}_A : \mathbb{R}^m \mapsto \mathbb{R}^n$ 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], [29] with very high probability. Yet, if the matrix $A^{\pm}$ is defined as $A^{\pm}_{j,k} = \text{sign}(A_{j,k})$, then $\mathcal{L}_A(x) = A^{\pm}x$ is also known to work with very high probability [30]. 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 [31]–[34], and suitably designed matrices are able to increase compression considerably compared 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 $\mathcal{O}(n \cdot m \cdot N/n) = \mathcal{O}(n \cdot N/\text{CR})$. However, 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.

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 these high-level reasons, short windows may benefit the implementation of the encoder also at a more physical level. In purely digital realizations [35]–[37], 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], [38]–[40], $y = A^{\pm}x$ is computed component-wise as $y_j = \sum_{k=0}^{n-1} A^{\pm}_{j,k} x_k$, i.e., accumulating the signal samples in the analog domain. This operation implies an analog storage to hold the intermediate sum value. However, independently of the actual implementation and technology, the approach is doomed to suffer from leakage and disturbance [9], [41]. 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 is expected to increase. Since κ affects m, any reduction of n tends to impair the compression ratio. As a remark, the trend with which κ/n increases when n decreases is a feature of the class of signals considered.

To get a quantitative feeling of these trends, we show in Fig. 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 [42]. 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 [43].

The value of κ is seen as a system parameter estimated at design-time so that the representations along the sparsity basis of most of the signal instances, feature a number of non-negligible elements not larger than κ. In Fig. 2, κ is estimated as the least number of entries of the sparse representation that includes 99.5% of the energy in the 99% of the ECG instances,

[2]Terms like *decoding* or *recovery* are also used to described this stage.

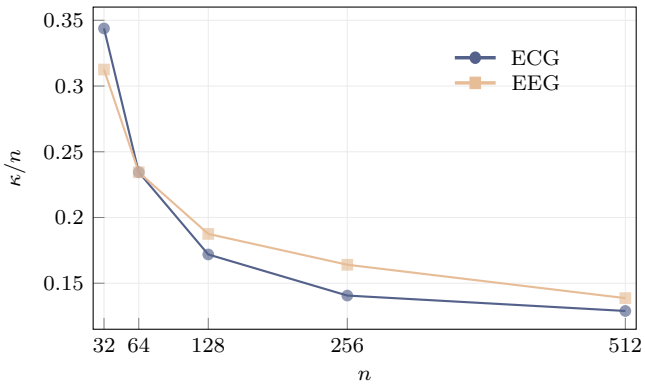


Fig. 2. The effect of reducing n on the normalized sparsity κ/n in the two examples of a synthetic ECG and a synthetic EEG signal.

and 95% of the energy in the 99% of the EEG instances[3]. 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.

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* $\operatorname{supp}\xi$ 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. However, 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 Fig. 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

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

where $\hat{x} = S\hat{\xi}$ is the reconstructed signal, $\|v\|_p$ indicates the p-norm of the generic vector v, and $\tau \geqslant 0$ 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 essential ingredient in BP and BPDN and is embedded in the 1-norm used in the objective function. The reason behind the use of the 1-norm minimization is to replace the minimization of the cardinality of the support of ξ that would yield a combinatorial problem. In fact, 1-norm minimization tends to select the ξ with the least number of non-zero entries among all the possible ξ satisfying the constraint [1]. This property is so critical 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. Note that, despite its fundamental merit, the 1-norm minimization is only a *proxy* of support identification, which works under suitable assumptions that are not necessarily satisfied in practice, especially for large κ/n values [1].

Since we enlarge the application of the CS framework to the cases where κ/n is quite large, 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 $\xi_{|s}$.

A. Oracle structure and training

The oracle we propose is based on a DNN trained on signals with the same statistical features of the one to be acquired. The DNN $\mathcal{N}_C : \mathbb{R}^m \mapsto [0,1]^n$ is 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 $(1+e^{-a})^{-1}$. Training also adapts the matrix A, so that encoder and decoder are jointly optimized to improve support identification and thus to improve reconstruction performance.

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 $t = 0,\ldots,T-1$) and of corresponding binary vectors $s^{(t)}$. The true support of $\xi^{(t)}$ encoded in $s^{(t)}$ and the output $o^{(t)} = \mathcal{N}_C\left(\mathcal{L}_A\left(x^{(t)}\right)\right)$ of the DNN are compared by means

[3]For 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.

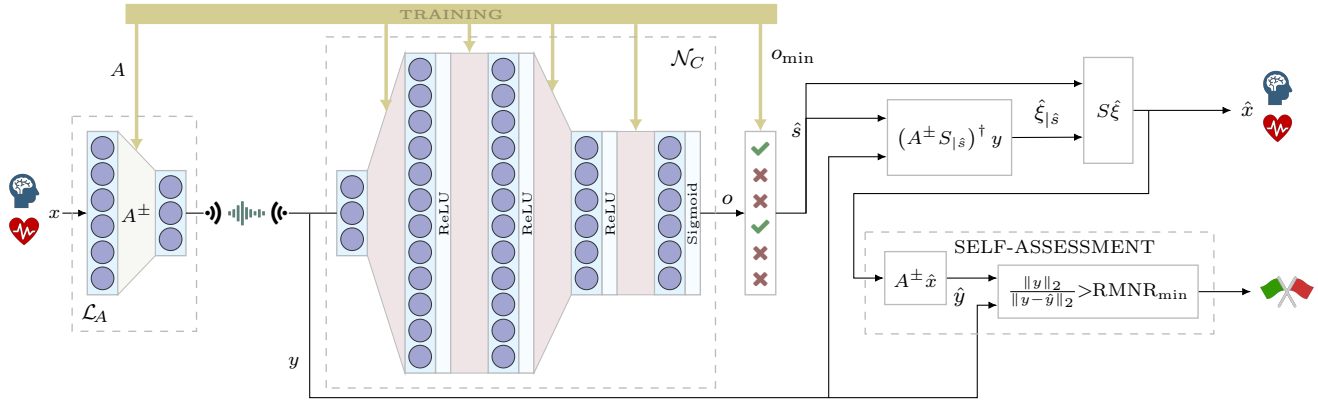


Fig. 3. Trained CS with support oracle block scheme including self-assessment capability. The fully connected DNN in the left is $\mathcal{N}_C \circ \mathcal{L}_A$ with an output layer $o \in [0,1]^n$. Estimated support $\hat{s}$ is such that $\hat{s}_j = 1$ if $o_j \geqslant o_{\min}$ and $\hat{s}_j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

of a loss function, which is a total component-wise clipped cross-entropy between s and o defined by

$$X^{(t)} = -\sum_{j|s_j^{(t)}=1} \mathrm{L}_\epsilon\left(o_j^{(t)}\right) - \sum_{j|s_j^{(t)}=0} \mathrm{L}_\epsilon\left(1-o_j^{(t)}\right) \tag{2}$$

where $\mathrm{L}_\epsilon(\cdot) = \min\{\log_2(1-\epsilon), \max\{\log_2(\epsilon), \log_2(\cdot)\}\}$ 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 $A^{\pm}_{j,k} = \mathrm{sign}(A_{j,k})$ 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 of 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 using the TensorFlow framework [44] with the help of the high-level API provided by Keras [45]. Training is performed with stochastic gradient descent, where each gradient step is computed with a mini-batch comprising of 30 signal instances and an initial learning rate value of 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 epochs[4]. For $n = 128$ and with m ranging in $[24, 64]$, the model counts from 124672 to 140032 parameters and in our examples is trained for 1000 epochs. Even assuming that each parameter is encoded in 4B, the total memory footprint is limited below 150KiB for $n = 64$ and below 550KiB for $n = 128$. Such requirements may easily fit within the memory budget of commercially available devices used for small scale computation and gateway tasks.

B. Performance indexes

The encoder-decoder chain may simultaneously perform more than one useful operation on the signal (see, e.g., [46]–[49] 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.

However, such 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 using the Reconstruction Signal-to-Noise Ratio (RSNR) defined as

$$\mathrm{RSNR} = \frac{\|x\|_2}{\|x-\hat{x}\|_2}\mathrm{dB} \tag{3}$$

where for any scalar a, the $a\,\mathrm{dB}$ 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 $t = 0, \ldots, T-1$) of signal instances recovered as $\hat{x}^{(t)}$. The first is the Average RSNR (ARSNR)

$$\mathrm{ARSNR} = \frac{1}{T}\sum_{t=0}^{T-1} \mathrm{RSNR}^{(t)} \tag{4}$$

while the second is the Probability of Correct Reconstruction (PCR) that, given a $\mathrm{RSNR}_{\min}$ value, is defined as

$$\mathrm{PCR} = \frac{1}{T}\#\left\{t \,\middle|\, \mathrm{RSNR}^{(t)} \geqslant \mathrm{RSNR}_{\min}\right\} \tag{5}$$

where $\#$ counts the number of elements in the set. The value of $\mathrm{RSNR}_{\min}$ has to be set accordingly to the minimum RSNR level that is considered *sufficient* for a correct reconstruction.

V. Trained CS with Support Oracle

The trained oracle can be exploited in the definition of the decoder reported in Fig. 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 $o_j \geqslant o_{\min}$ and $\hat{s}_j = 0$ otherwise. Starting from $\hat{s}$ we finally estimate

$$\hat{\xi}_{|\hat{s}} = \left(A^{\pm}S_{|\hat{s}}\right)^{\dagger} y \tag{6}$$

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

[4]In each epoch the training algorithm walks through the entire training set.

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.

The sensing matrix design follows the rakeness-based CS framework [33], [34] that we have verified to yield better results compared to the classical independent assignment of ± 1 to each of the entries of $A^{\pm}$. Performance improvement comes from the adaptation of the statistics of the rows of the sensing matrix to the statistics of the acquired class of signals. As decoders, we consider BP and BPDN as presented in (1) along with Orthogonal Matching Pursuit (OMP) [15] and Generalized Approximate Message Passing (GAMP) [14]. OMP is a lightweight greedy approach that iteratively estimates the signal support while GAMP is often better than BP and BPDN as it exploits Gaussian approximation of BP that usually holds for large n values. When dealing with ECGs, we also test the performance of the Weighted ℓ_1 minimization (WL1) [18] as a representative of decoders that exploit statistical priors on the signal support. In all the tested cases, BP outperforms BPDN such that, in the rest of the paper, we consider BP as a reference for standard CS decoder.

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 resulting in an Intrinsic Signal-to-Noise Ratio (ISNR) equal to $60\,\mathrm{dB}$. The achieved performances are reported in Fig. 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 $\mathrm{CR} = 1.6$ up to $\mathrm{CR} = 4$. TCSSO outperforms all other techniques and allows us to work at compression ratios much larger than those commonly achievable, while still requiring a limited computational effort since $n = 64$. For example, to guarantee ARSNR $= 50\,\mathrm{dB}$, results in Fig. 4(a),(c) show that by using TCSSO one may get $\mathrm{CR} \approx 3.5$ for ECGs and ≈ 2.9 for EEGs. In the same setting, RAK+WL1 is the best performing competitor for ECGs with $\mathrm{CR} \approx 2.2$ while RAK+BP is to be considered a benchmark for the EEGs with $\mathrm{CR} \approx 1.8$.

Fig. 5 shows how the situation changes when n increases from 64 to 128. Performance is reported only in terms of ARSNR and for TCSSO along with its best competitor. The increase of n positively impacts performances in general since κ/n decreases. Nevertheless, TCSSO still outperforms the best of the traditional CS frameworks. Considering ARSNR $= 50\,\mathrm{dB}$ as the desired quality of service, TCSSO works with $\mathrm{CR} \approx 4.4$ and $\mathrm{CR} \approx 2.9$ while the competitors give at most $\mathrm{CR} \approx 2.7$ and $\mathrm{CR} \approx 2.2$ in case of ECGs and EEGs respectively.

A. Preliminary evidence on real signals

The ideal path for applying our method in real-world cases is to collect enough acquisitions to substantiate both a training and a validation set. The acquisitions to which we have access do not currently allow such a thorough assessment. However, some evidence can be given by using synthetic data for the learning phase, and real-world signals for a preliminary assessment. Such an approach is suboptimal since there is no guaranteed coherence between the training set used for learning and the validation set used for assessment. The results are still encouraging.

In particular, we may consider the waveforms contained in the MIT database for testing compression of ECG signals [50], [51] and the pool of acquisitions used in [43] for EEG. A sample comparison between original and reconstructed waveforms for $n = 64$ and $m = 32$ (which is equal to 2κ) is reported in Fig. 6. Despite the appearance of some artifacts introduced by the decoder, the plots show that, even with the suboptimal setting, our method is able to yield acceptable reconstructions with extremely small n and with m below the classical threshold 2κ.

VI. 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 $y = A^{\pm}S_{|s}\xi_{|s}$ implies that $y \in \mathrm{span}\left(A^{\pm}S_{|\hat{s}}\right)$, where, for any matrix M, $\mathrm{span}\,(M)$ is the subspace generated by the linear combination of its columns. This has a twofold consequence: *i*) (6) computes $\hat{\xi}_{|\hat{s}} = \xi_{|s}$, *ii*) if $\hat{\xi}$ is mapped back we have $A^{\pm}S\hat{\xi} = y$.

However, 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 $y \notin \mathrm{span}\left(A^{\pm}S_{|\hat{s}}\right)$. This has a twofold consequence: i) (6) 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. Nevertheless, it may map $\hat{\xi}$ back to measurement obtaining $\hat{y} = A^{\pm}S_{|\hat{s}}\left(A^{\pm}S_{|\hat{s}}\right)^{\dagger}y = A^{\pm}\hat{x}$ 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

$$\mathrm{RMNR} = \frac{\|y\|_2}{\|y - \hat{y}\|_2} \tag{7}$$

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

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

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

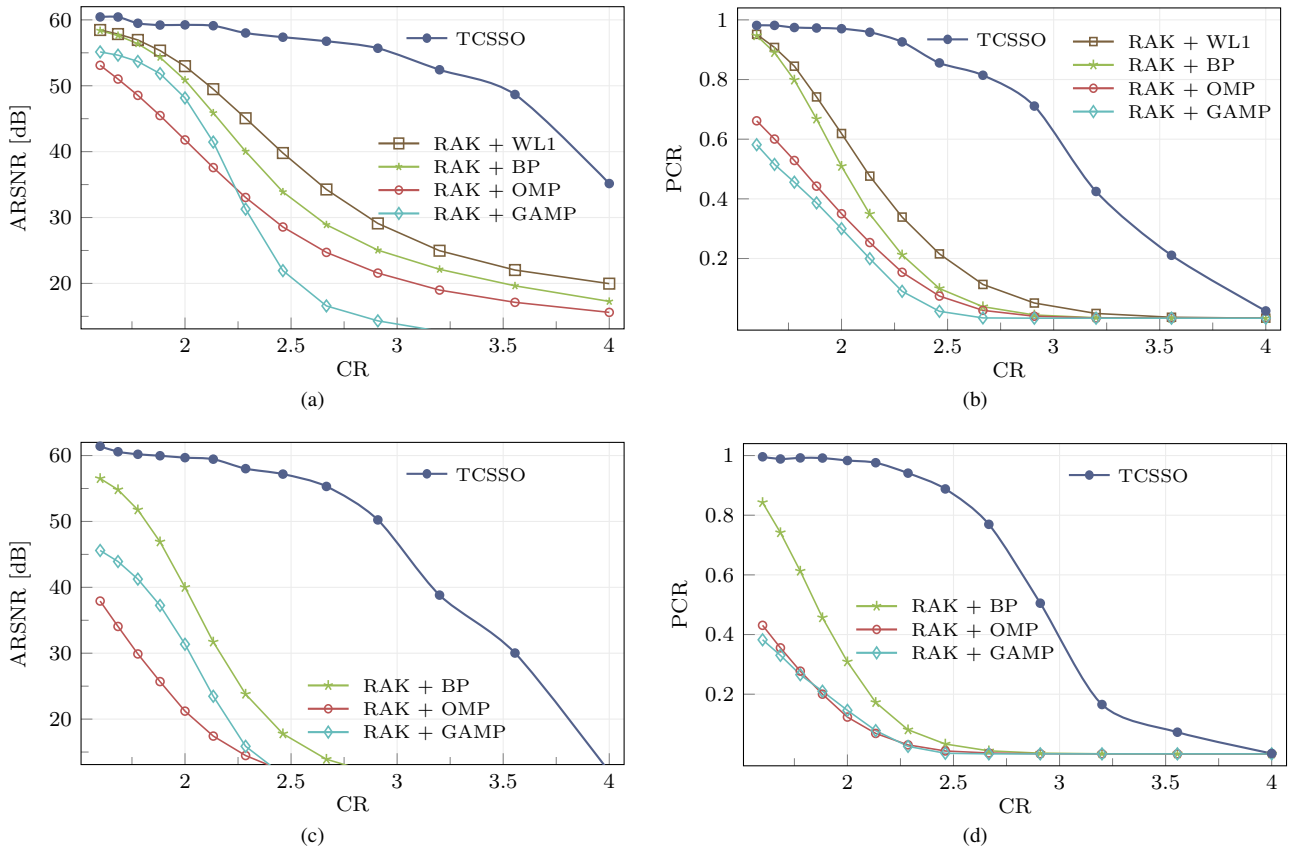


Fig. 4. Reconstruction performance for ECG (a)-(b) and EEG signals (c)-(d) in terms of both ARSNR (a)-(c) and PCR (b)-(d) with $\mathrm{RSNR}_{\min} = 55\,\mathrm{dB}$. The support oracle decoder with trained sensing matrices, TCSSO, is compared against OMP, BP, and GAMP with adapted sensing matrices (rakeness-based CS, RAK). For the ECG, we also consider the adapted decoder WL1 in [18].

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 Fig. 7 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. However, recovery is theoretically possible as no two images $\iota_{j,k}$ and $\iota_{j',k'}$ are the same. Therefore 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 $y \in \iota_{0,1}$. 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 (6), the vector y is mapped back to $\hat{\xi}$ on that plane, which 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, Fig. 8 reports some Montecarlo evidence on the relationship between RMNR and RSNR for the ECG signals and in three different configurations. In Fig. 8a no noise is present and $m = 32 = 2\kappa$; in Fig. 8b ISNR $= 60\,\mathrm{dB}$ and $m = 32 = 2\kappa$, whereas in Fig. 8c 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 ($\hat{s}_j \geqslant s_j$ for all $j = 0, \ldots, n-1$, 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 $j = 0, \ldots, n-1$, 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 $\mathrm{RMNR} \geqslant \mathrm{RMNR}_{\min}$ and for a negative event otherwise. As the threshold $\mathrm{RMNR}_{\min}$ 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 $\mathrm{RMNR}_{\min}$ even good reconstructions can be declared incorrect.

The ideal conditions in Fig. 8a result in perfect self-assessment capabilities. When noise is added as in Fig. 8b, positive and negative cases get mixed but remain identifiable by looking at RMNR.

Though no noise is present in Fig. 8c, the fact that $m < 2\kappa$ makes the number of measurements insufficient for signal

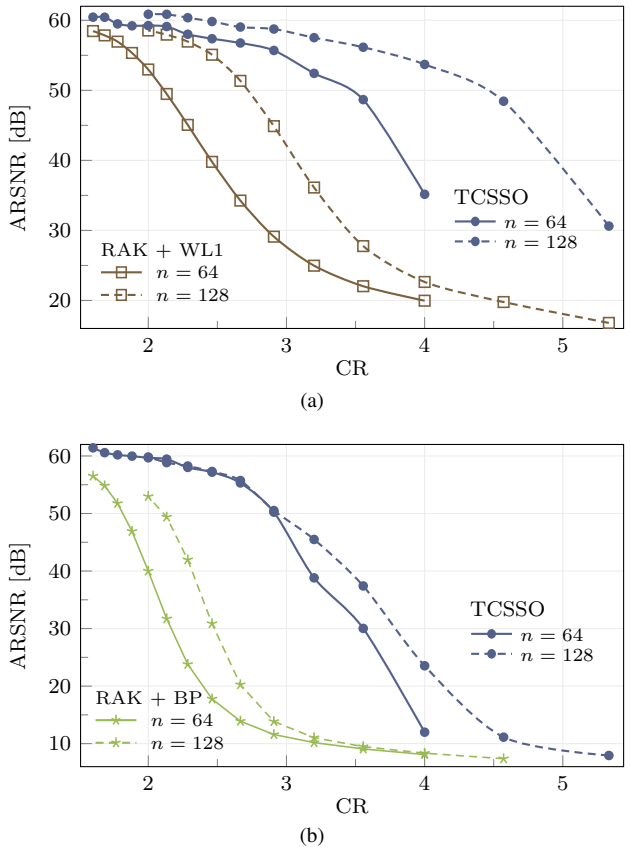


Fig. 5. Performance in terms of ARSNR ECG (a) and EEG signals (b) as a function of the compression ratio for both TCSSO and the best observed traditional approach (RAK + WL1 for ECG while the EEG case refers to RAK + BP). Results are for $n = 64$ as well as for $n = 128$.

reconstruction, as there is no guarantee that only one κ-sparse signal ξ corresponds to the given y through $A^{\pm}S$. Hence, more than one support corresponding to the measurement exists. 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 $o_{\min}$) 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 Fig. 8c). However, pseudo-inversion 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 Fig. 8c), 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 $\mathrm{RMNR}_{\min}$, 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.

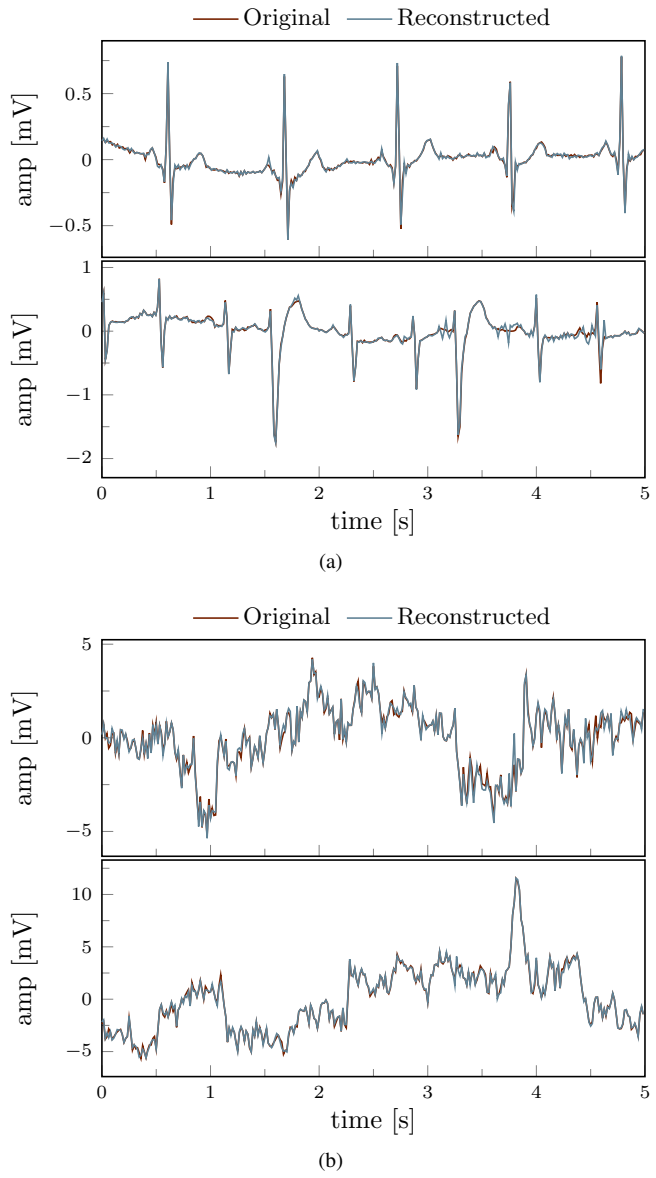


Fig. 6. Comparisons between real-world signals and their corresponding waveforms reconstructed by TCSSO for ECGs (a) and EEGs (b). The top plot (a) refers to records 11950_03 and 12531_03 taken from the online repository MIT-BIH ECG compression test database [50]. The bottom plot (b) contains EEG signal chunks of the "Fz" electrode from the pool of acquisition described in [43]. For both cases we adopt $n = 64$ and $m = 32$.

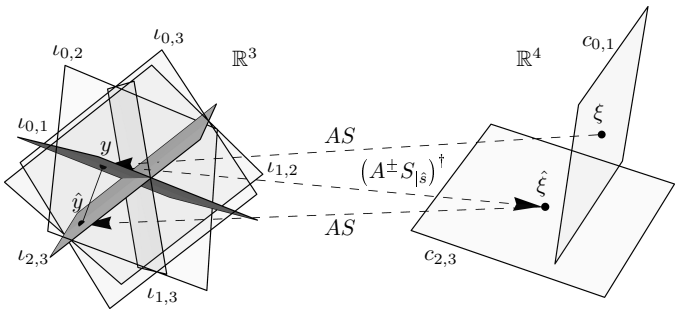


Fig. 7. The mechanism granting self-assessment capabilities to decoders based on a support oracle

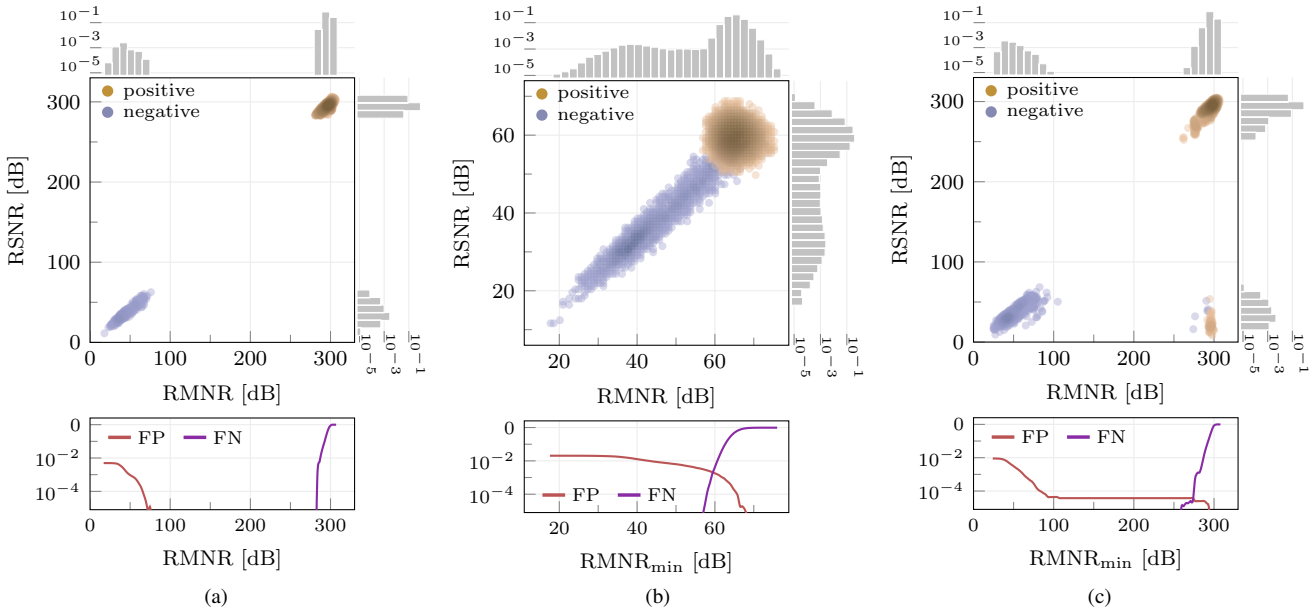


Fig. 8. The relationship between RSNR and RMNR for ECG signals. In all cases $n = 64$ and $\kappa = 16$. Orange dots correspond to cases in which $\hat{s}$ includes all the components of s, while blue dots correspond to $\hat{s}$ failing to identify some components in s. Above and to the right of the scatter plots are logarithmic histograms estimating the probability density of RMNR and RSNR. In (a) $m = 32 = 2\kappa$ and ISNR $= \infty$. In (b) $m = 32 = 2\kappa$ and ISNR $= 60$ dB. In (c), $m = 24 < 2\kappa$ and ISNR $= \infty$.

In general, the value of $\mathrm{RMNR}_{\min}$ can be decided once that $o_{\min}$ is set, by a further pass over the training set. This allows us to estimate false positive and false negative curves as in Fig. 8, and use them as criteria. In the following, we will set $\mathrm{RMNR}_{\min}$ as the largest value for which false negative probability is negligible. Whenever a failure is detected, the decoder may take different actions whose effectiveness depends on the final applications.

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

i) raising a warning and mark the current window as potentially incorrect;
ii) feeding the warning back to the encoder and require further information to correct the reconstruction (thus lowering the CR for this instance);
iii) triggering another decoder on the same measurement vector hoping that this will improve reconstruction;
iv) any combination of the above.

As a partial and 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 GAMP[5] as a second-wind decoder.

Fig. 9 plots the probability that GAMP yields a RSNR larger than what is given by TCSSO when applied to the instances that the latter marks as incorrectly recovered as $\mathrm{RMNR} < \mathrm{RMNR}_{\min}$, 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 $\mathrm{CR} \leqslant 2$.

[5]GAMP 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.

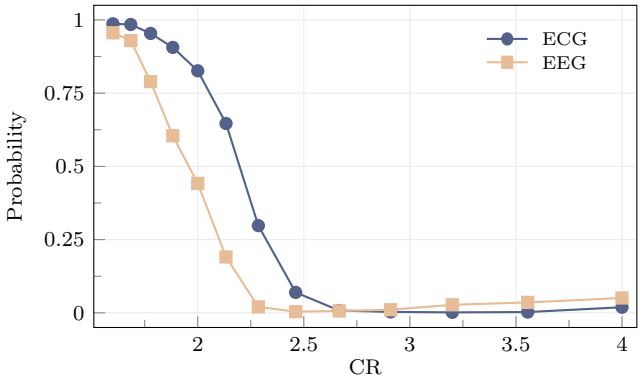


Fig. 9. Probability for GAMP decoder to reconstruct both ECG and EEG signals with RSNR higher than TCSSO in case of TCSSO failure. $\mathrm{Prob}\left(\mathrm{RSNR}_{\mathrm{GAMP}} > \mathrm{RSNR}_{\mathrm{TCSSO}} \,|\, \mathrm{RMNR} < \mathrm{RMNR}_{\min}\right)$.

VII. 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.

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

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 $nm = n^2\mathrm{CR}^{-1}$ thus yielding $n\mathrm{CR}^{-1}$ AC/sample. Further to time-complexity, the memory footprint is dominated by the storage of the matrix $A^{\pm}$, and requires a number of entries equal to $nm = n^2\mathrm{CR}^{-1}$. In principle, matrix entries are bits. However, 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., [19]), this is why we express the memory footprint as the number of entries in $A^{\pm}$.

From the blue curves in Fig. 5, one gets that a 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 of reconstruction quality, complexity, and memory footprint for ECG signals when n goes from 64 to 128, with CR ranging from 2 to 4. At high CR levels, an increase in terms of ARSNR (e.g., with $\mathrm{CR} = 4$, +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. However, for lower compression ratios, the increase in resource needs is not justified by the limited increase in performance: for $\mathrm{CR} = 2$, memory footprint and complexity 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 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, starting from the fact that the number of MAC operations required in a fully connected layer with n nodes, each with i inputs, is ni.

Neglecting the input layer, that has m nodes and that requires no operations, the oracle $\mathcal{N}_C$ is composed by 3 fully connected hidden layers with $2n$, $2n$ and n nodes, and a final fully connected output layer with n nodes. The number of inputs of these layers is therefore m, $2n$, $2n$ and n, respectively. The layer-by-layer number of MACs required for the forward pass is $2nm$, $4n^2$, $2n^2$ and n^2, giving rise to a total of $(2m+7n)n = (2/\mathrm{CR}+7)n^2$ MACs for each window thus yielding $(2/\mathrm{CR}+7)n$ MAC/sample.

After support estimation, additional MACs are needed to compute $\hat{x}$. In particular, we focus on the computational cost of $\hat{\xi} = B^{\dagger}y$, with $B^{\dagger}$ the Moore-Penrose pseudoinverse of $B = A^{\pm}S_{|\hat{s}}$, i.e., of a matrix with m rows and a number c of columns $\kappa \leqslant c \leqslant n$, with $c \simeq \kappa$ being the most frequent case. The computational complexity of pseudo-inversion reflects its analytical formula such that $\hat{\xi} = B^{\top}\left(BB^{\top}\right)^{-1}y$ must be computed. Since B is a $m \times c$ matrix, $BB^{\top}$ requires m^2c MACs, and the inversion entails $2m^3$ MACs. Now, the right-multiplication $\left(BB^{\top}\right)^{-1}$ by y costs m^2 MACs and the final left-multiplication results in mc MACs. Considering all contributions, we arrive at estimating a total of $m(2m^2+m\kappa+m+\kappa)$ for the typical $c = \kappa$ case. The complexity is then equal to $(2n\mathrm{CR}^{-2}+n\kappa/n\mathrm{CR}^{-1}+\kappa/n+\mathrm{CR}^{-1})n\mathrm{CR}^{-1}$ MAC/sample.

We may compare the complexity of TCSSO decoding with that of OMP, which is known to be one of the most light-weighted approaches. We consider the standard implementation of OMP as described in [52]. A modified version (bWOMP) of this algorithm has been proposed in [19] to exploit the same statistical prior described in [18] and improve reconstruction performance with no significant increase of computational complexity. The detailed description of OMP is out of the scope of this paper, and we refer to [52] for details. Knowing that OMP is an iterative algorithm that estimates the signal support in at least κ iteration, we limit ourselves to provide the complexity in terms of the number of MAC for the j-th iteration that is $nm + 2m(j-1) + 2m + 2jm$. This yields a total of at least $2\kappa m + 2\kappa^2 m + \kappa nm$ MACs. After that, OMP computes the pseudo-inverse of a matrix of the same size as the $B = A^{\pm}S_{|\hat{s}}$ in TCSSO. The total complexity of the iterative part is therefore given by $(2 + 2n\,\kappa/n + n)n\kappa/n\mathrm{CR}^{-1}$ MAC/sample and must be compared with the computational effort required by the oracle that is $(2/\mathrm{CR}+7)n$ MAC/sample.

Though the contributions to the computational complexities computed above have different 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 $\mathrm{CR} = 2$ (one of our ECG cases) the first part of OMP entails some 784 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 $\mathrm{CR} = 4$ (one of our EEG cases) the first part of OMP entails some 1183 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, showing that, though TCSSO allows implementing extremely lightweight encoders, the decoder does not have to compensate by increasing its computational require-

ments compared to conventional decoders. As a consequence, since bWOMP has complexity similar to OMP, at least in the settings we analyzed, the complexity of TCCSO is comparable to the one of decoders that use a statistical prior on the signal support.

VIII. 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, and allows the application of 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 is 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 above 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 cases, 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 generator[6] of ECGs is thoroughly discussed in [53]. 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 $60\,\mathrm{dB}$.

The setup is the same detailed in [33]. The heart-beat rate is randomly set using an uniform distribution between $60\,\mathrm{beat/min}$ and $100\,\mathrm{beat/min}$. We generate chunks of $2\,\mathrm{s}$ with a $256\,\mathrm{sample/s}$ 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 between a training set and a 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 signal[7] can be found in [54]. 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 the 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 can 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 $1024\,\mathrm{sample/s}$ with a stimulus frequency of $1\,\mathrm{Hz}$.

We generate tracks corresponding to 50 different patients by starting from the parameters used in [54] 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, $\pm 0.05\,\mathrm{Hz}$ 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, $\pm 1\,\mathrm{Hz}$ for the peak frequency, and ± 4 for the peak amplitude.

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

[6]The MATLAB code is freely available for downloaded from the Physionet website at http://physionet.org/content/ecgsyn/

[7]The 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-eeg-data-generator

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.
[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. Jaspan, R. Fleysher, 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] L. F. Polanía, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, "Exploiting prior knowledge in compressed sensing wireless ecg systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 508–519, March 2015.

[18] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, "Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted ℓ_1 minimization reconstruction," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 520–528, March 2015.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] M. Elad, "Optimized projections for compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5695–5702, 2007.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, 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.

[41] 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.

[42] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Access Online via Elsevier, 2008.

[43] 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.

[44] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[45] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[46] 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.

[47] 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.

[48] 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.

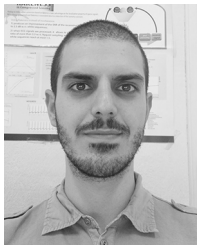
[49] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[50] M. G. B. *et al.*, "Evaluation of the "trim" ecg data compressor," *Computers in Cardiology*, no. 15, pp. 167–170, 1988.
[51] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, Jun. 2000.
[52] F. Pareschi, M. Mangia, D. Bortolotti, A. Bartolini, L. Benini, R. Rovatti, and G. Setti, "Energy analysis of decoders for rakeness-based compressed sensing of ecg signals," *IEEE Transactions on Biomedical Circuits and Systems*, vol. PP, no. 99, pp. 1–12, 2017.
[53] 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.
[54] 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.

Mauro Mangia (S'09-M'13) eceived the B.Sc. and M.Sc. degrees in electronic engineering and the Ph.D. degree in information technology from the University of Bologna, Bologna, Italy, in 2005, 2009, and 2013, respectively. He was a Visiting Ph.D. Student with the Ecole Polytechnique Federale de Lausanne in 2009 and 2012. He is currently a Post-Doctoral Researcher with ARCES, Statistical Signal Processing Group, University of Bologna. His research interests are in nonlinear systems, machine learning, compressed sensing, anomaly detection, Internet of Things, Big Data analytics and optimization.

He was a recipient of the 2013 IEEE CAS Society Guillemin-Cauer Award and of the 2019 IEEE BioCAS Transactions Best Paper Award. He received the Best Student Paper Award at ISCAS2011. He was the Web and Social Media Chair for ISCAS2018.

Luciano Prono received the M.Sc. degree in Electronics Engineering in 2019 from Politecnico di Torino, where he is currently pursuing the Ph.D. degree in Electrical, Electronics and Communication Engineering. His main research interests are compressed sensing, low power deep learning systems and neuromorphic systems.



Alex Marchioni received the B.S. and M.S. degree (with honors) in electronic engineering from the University of Bologna, respectively in 2011 and 2015. In 2018, he joined the Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi" (DEI) of the University of Bologna, where he is currently working as research fellow. His research interests include compressed sensing, biomedical applications and signal processing for the Internet of Things and Big Data analytics.

Fabio Pareschi (S'05-M'08-SM'19) received the Dr. Eng. degree (Hons.) in electronic engineering from the University of Ferrara, Italy, in 2001, and the Ph.D. degree in information technology from the University of Bologna, Italy, in 2007, under the European Doctorate Project (EDITH).

He is currently an Assistant Professor with the Department of Electronic and Telecommunication, Politecnico di Torino. He is also a Faculty Member with ARCES, University of Bologna. His research activity focuses on analog and mixed-mode electronic circuit design, statistical signal processing, compressed sensing, random number generation and testing, and electromagnetic compatibility.

Dr. Pareschi received the Best Paper Award at ECCTD 2005 and the Best Student Paper Award at EMC Zurich 2005 and IEEE EMCCompo 2019. He was a recipient of the 2019 IEEE BioCAS Transactions Best Paper Award. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-PART II from 2010 to 2013. He is currently Associate Editor for the IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS.

Riccardo Rovatti (M'99-SM'02-F'12) received the M.S. degree in electronic engineering and the Ph.D. degree in electronics, computer science, and telecommunications from the University of Bologna, Italy, in 1992 and 1996, respectively. He is currently a Full Professor of electronics with the University of Bologna. He has authored approximately 300 technical contributions to international conferences and journals and two volumes. His research focuses on mathematical and applicative aspects of statistical signal processing and on the application of statistics to nonlinear dynamical systems.

He was Distinguished Lecturer of the IEEE CAS Society for the years 2017-2018. He was a recipient of the 2004 IEEE CAS Society Darlington Award, the 2013 IEEE CAS Society Guillemin-Cauer Award and the 2019 IEEE BioCAS Transactions Best Paper Award. He received the Best Paper Award at ECCTD 2005 and the Best Student Paper Award at the EMC Zurich 2005 and ISCAS 2011. He contributed to nonlinear and statistical signal processing applied to electronic systems.

Gianluca Setti (S89,M91,SM02,F06) received a Ph.D. degree in Electronic Engineering and Computer Science from the University of Bologna in 1997.From 1997 to 2017 he has been with the School of Engineering at the University of Ferrara, Italy as an Assitant, Associate and, since 2009 as a Full Professor of Circuit Theory and Analog Electronics. Since December 2017 he is a Professor of Electronics for Signal and Data Processing at the Department of Electronics and Telecommunications (DET) of Politecnico di Torino, Italy. Since 2002 is also a permanent (in kind) faculty member of ARCES, University of Bologna. His research interests include nonlinear circuits, recurrent neural networks, statistical signal processing, electromagnetic compatibility, compressive sensing, biomedical circuist and systems, power electronics, design and implementation of IoT nodes.

Dr. Setti received the 1998 Caianiello prize for the best Italian Ph.D. thesis on Neural Networks. He is also receipient of the 2013 IEEE CAS Society Meritorious Service Award and co-recipient of the 2004 IEEE CAS Society Darlington Award, of the 2013 IEEE CAS Society Guillemin-Cauer Award, the 2019 IEEE Transactions on Biomedical Circuits and Systems best paper award, as well as of the best paper award at ECCTD2005, and the best student paper award at EMCZurich2005, ISCAS2011, PRIME2019 and EMCCOMPO 2019.

He held several editorial positions and served, in particular, as the Editor-in-Chief for the IEEE Transactions on Circuits and Systems - Part II (2006-2007) and of the IEEE Transactions on Circuits and Systems - Part I (2008-2009). He also served in the editorial Board of IEEE Access (2013-2015) and, since of the Proceedings of the IEEE (2015-2018). Since 2019 he served as the first non US Editor-in-Chief of the Proceedings of the IEEE, the flagship journal of the Institute.

Dr. Setti was the Technical Program Co-Chair of NDES2000 (Catania), ISCAS2007 (New Orleans), ISCAS2008 (Seattle), ICECS2012 (Seville), BioCAS2013 (Rotterdam) as well as the General Co-Chair of NOLTA2006 (Bologna) and ISCAS2018 (Florence).

He was a Distinguished Lecturer (2004-2005 and 2014-2015) of the IEEE CAS Society, as well as a member of its Board of Governors (2005-2008), and served as the 2010 CASS President. He held several other volunteer positions for the IEEE and in 2013-2014 he was the first non NorthAmerican Vice President of the IEEE for Publication Services and Products.