PAPER

# A fast algorithm for spatiotemporal signals recovery using arbitrary dictionaries with application to electrocardiographic imaging

**S F Caracciolo**[1,2,*], **C F Caiafa**[3], **F D Martínez Pería**[1,4] **and P D Arini**[1,2]

[1] Instituto Argentino de Matemática 'Alberto P. Calderón', CONICET, Saavedra 15 (C1083ACA), Ciudad Autónoma de Buenos Aires, Argentina
[2] Instituto de Ingeniería Biomédica, UBA, Av. Paseo Colón 850 (C1063ACV), Ciudad Autónoma de Buenos Aires, Argentina
[3] Instituto Argentino de Radioastronomía—CCT La Plata, CONICET / CIC-PBA / UNLP, (1894) V. Elisa, Buenos Aires, Argentina
[4] Centro de Matemática de La Plata, UNLP, 50 y 115 (B1900AJJ), La Plata, Buenos Aires, Argentina
[*] Author to whom any correspondence should be addressed.

**E-mail:** scaracciolo@conicet.gov.ar, ccaiafa@fi.uba.ar, francisco@mate.unlp.edu.ar and pedro.arini@conicet.gov.ar

**Keywords:** ECGI, group lasso, Kronecker product, sparse regularization

## Abstract

This paper presents a method to solve a linear regression problem subject to group *lasso* and ridge penalisation when the model has a Kronecker structure. This model was developed to solve the inverse problem of electrocardiography using sparse signal representation over a redundant dictionary or frame. The optimisation algorithm was performed using the block coordinate descent and proximal gradient descent methods. The explicit computation of the underlying Kronecker structure in the regression was avoided, reducing space and temporal complexity. We developed an algorithm that supports the use of arbitrary dictionaries to obtain solutions and allows a flexible group distribution.

## 1. Introduction

Generally, real-world signals exhibit spatiotemporal characteristics. Signals are measured by sensors at different spatial locations, recording the information with a given sampling rate. For example, electrocardiographic imaging (ECGI) is a technique used to measure the electrical activity in the thorax, and it then uses a model to estimate the electrical activity on the cardiac muscle surface. This model is based on the laws of electromagnetism which determine the spatiotemporal dynamics of the fields generated by sources. The family of problems that uses models to infer sources from measurements are called inverse problems.

Modelling all the details can produce a burden on the computation, making it unfeasible. Generally, ECGI models assume that the thorax and heart have static geometries with a homogeneous, isotropic, and free source medium [1]. The inverse problem in electrocardiography [2] depends on the cardiac sources under consideration, such as the dipoles [3], transmembrane voltage [4], extracellular potential [5] and activation and recovery models [6]. When a model is discretised by a boundary or finite element method, numerical errors can occur owing to the precision and meshing approximations [7–9].

The aforementioned limitations harm modelling and promote ill-conditioning. Thus, a well-conditioned model requires that the following properties are fulfilled: (1) the existence and uniqueness of the solution, and (2) the stability of solutions [10]. There are several approaches, called regularisations, to mitigate ill-conditioning, based on the inclusion of *a priori* information. For instance, in the truncated singular value decomposition approach, the idea is to set the small singular values of the model to zero, to improve the conditioning [11]. In fact, *a priori* information does not depend on the application; instead, it is based completely on the stability of the problem.

However, ECGI regularisation methods can be formulated from a statistical perspective [12, 13]. These schemes allow the inclusion of prior information from training sets of either simulated or real data. But a good estimation requires a large amount of information, which is an important limitation of ECGI.

In this study, we are particularly interested in regularisation by norm constraints on the solutions. The idea is to include *a priori* information as assumptions expressed through the norm of the vectors. For instance, Tikhonov spatial regularisation is a $\ell_2$-norm constraint over some operator applied to the solutions, such as identity (zero order), discrete gradient (first-order) or Laplacian (second-order) operators. These methods assume that the operator solution product is dense and are regarded as the gold standard in most ECGI studies [5, 14]. However, a $\ell_1$-norm constraint called *lasso* (Least Absolute Shrinkage and Selection Operator) [15], promoted sparse solutions. In [16] the authors hypothesised that the normal derivative of the cardiac potential is sparse, which justifies the use of the $\ell_1$ penalty. Additionally, the $\ell_1$-norm was tested on data terms to diminish the effect of outliers on epicardial potential reconstruction [17]. Both norms and mixtures of them are broadly used in convex optimisation, for example, *elastic-net* [18], group *lasso* [19] and sparse group *lasso* [20]. Except for [21], these methods have rarely been applied to ECGI.

Some attempts have been made to incorporate *a priori* information that is more closely connected to the application. In [22, 23], a spatial regularisation approach was developed to simulate a realistic spatial basis for the heart potential to promote sparsity on the decomposition coefficients, and then use the $\ell_1$-norm constraint.

In [24], the authors used the same constraint, but a temporal basis was defined by orthogonal wavelets to generate sparse coefficients of cardiac signals. Finally, [21] used temporal regularisation based on orthogonal wavelets. But also includes the concept of group sparsity to add spatial information to the regularisation. To make this, a group *lasso* plus a ridge regression constraint was implemented under two conditions: (1) the wavelet transform must be generated by a tight frame [25] and (2) each group includes all spatial nodes.

In this paper, we present an algorithm to extend the framework of [21] to support any dictionary structure, that is, not only tight frames, and to allow more flexible group distributions.

## 2. Materials and methods

### 2.1. Problem formulation

In this study, we deal with linear models discretised over space at $p$-nodes (measurements) and $q$-nodes (sources) denoted by $\mathbf{A} \in \mathbf{R}^{p\times q}$. If the model is static over time, the relation between the variables of interest can be represented by a matrix operation along $n$ samples:

$$\mathbf{Y} = \mathbf{AX} + \mathbf{H}, \tag{1}$$

where $\mathbf{Y} \in \mathbf{R}^{p\times n}$, $\mathbf{X} \in \mathbf{R}^{q\times n}$, and $\mathbf{H} \in \mathbf{R}^{p\times n}$ are matrices with measurements, sources and noise, respectively.

To design a flexible regularization framework, we propose to decompose the solutions using an arbitrary frame or dictionary to penalise the decomposition coefficients. Each row of $\mathbf{X}$ is a temporal signal that can be decomposed into a weighted sum of $k$ atoms. Therefore, a synthesis operator or dictionary $\mathbf{D} \in \mathbf{R}^{n\times k}$ with atoms per column computes the signals as

$$\mathbf{X} = \mathbf{\Theta}\mathbf{D}^T, \tag{2}$$

where $\mathbf{\Theta} \in \mathbf{R}^{q\times k}$ is the coefficients matrix. Replacing (2) into (1), using the Kronecker product property $(\mathbf{X}_2 \otimes \mathbf{X}_1)v = \mathbf{vec}(\mathbf{X}_1\mathbf{V}\mathbf{X}_2^T)$ and denoting $v := \mathbf{vec}(\mathbf{V})$ the vectorize operator which stacks vertically the columns of $\mathbf{V}$, we get,

$$y = \overbrace{(\mathbf{D} \otimes \mathbf{A})}^{\mathbf{Z}}\theta + h, \tag{3}$$

where $\mathbf{Z} \in \mathbf{R}^{pn\times qk}$ has a Kronecker structure. We highlight that the explicit computation of $\mathbf{Z}$ becomes impractical and should be avoided. For example, if $p = 200$, $q = 2000$, $n = 500$ and $k = 1000$ the Kronecker product requires ~745 Gb using a single-precision datatype, whereas $\mathbf{A}$ and $\mathbf{D}$ require ~1.5 Mb and ~1.9 Mb, respectively. This restricts us to develop a solver that avoids the explicit computation of $\mathbf{Z}$.

The existence and uniqueness of solutions to system (3) depend on the shape and rank of $\mathbf{Z}$. However, we focus on unstable models in which a cost function and a constraint are mandatory to obtain a unique and stable approximated solution.

By assuming that the error is independent and identically Gaussian distributed, we set up an ordinary least squares problem:

$$\min_{\theta} \frac{1}{2N}\|\mathbf{Z}\theta - y\|_2^2, \tag{4}$$

being $N = p$–$n$. Similar to [21], the constraint is a mixing between the $\ell_2$ and $\ell_{2,1}$ norms, depending on a parameter $\alpha \in [0, 1]$,

$$(1-\alpha)\frac{1}{2}\|\theta\|_2^2 + \alpha\sum_{i=0}^{m-1}\eta_i\|\theta_{\gamma_i}\|_2 \leqslant \epsilon. \tag{5}$$

If $\alpha = 0$ the constraint reduces to Tikhonov regularisation, whereas if $\alpha = 1$, then a group *lasso* is obtained. We define $\Gamma$ as a partition of $\Phi := \{0,\ldots, qk-1\}$, that is, each group or set in $\Gamma$ is non-empty, pairwise disjoint, and the union of the sets in $\Gamma$ covers $\Phi$. We call $m := \mathbf{card}(\Gamma)$ the number of groups, and we refer to each set in $\Gamma$ as $\gamma_i$, where $i \in \{0,\ldots,m-1\}$. Additionally, we define $x_{\gamma_i} := x[\gamma_i]$, $\mathbf{X}_{\gamma_i} := \mathbf{X}[:,\gamma_i]$, and $X_{\gamma_i} := \mathbf{X}[:,\gamma_i]$ if $\mathbf{card}(\gamma_i) = 1$. The constant $\eta_i > 0$ is used to weight the norm $\ell_{2,1}$ when the groups have different sizes, i.e., $\mathbf{card}(\gamma_i) \neq \mathbf{card}(\gamma_j)$.

Finally, rewriting (4) in terms of groups and using a Lagrange formulation of (4) and (5), we introduce a hyperparameter $\lambda > 0$ to control the penalization weight,

$$\mathop{\mathbf{min}}_{\theta} \frac{1}{2N}\left\|\sum_{i=0}^{m-1}\mathbf{Z}_{\gamma_i}\theta_{\gamma_i} - y\right\|_2^2 + \lambda P_\alpha(\theta) \tag{6}$$

$$P_\alpha(\theta) = (1-\alpha)\frac{1}{2}\|\theta\|_2^2 + \alpha\sum_{i=0}^{m-1}\eta_i\|\theta_{\gamma_i}\|_2 \tag{7}$$

### 2.2. Optimization strategy

The strategy to solve a group *lasso* problem was first presented in [19] and extended to the non-orthogonal $(\mathbf{Z}_{\gamma_i}^T\mathbf{Z}_{\gamma_i} \neq I)$ case in [20]. We use the former approach which involves a block coordinate descent algorithm nested using the proximal gradient method. We made slight adaptations to incorporate the $\ell_2$ term and we added some constraints into the partition $\Gamma$ in order to exploit the Kronecker structure of $\mathbf{Z}$.

### *2.2.1. Block coordinate descent (BCD)*

The function to be minimized in (6) is convex and non-differentiable due to the $\ell_2$ norm involved in the group *lasso* term. However, a separability condition is fulfilled over $\theta_{\gamma_i}$ coordinates in the nondifferentiable part, which ensures that the BCD algorithm converges to a global minimum [26, 27].

The BCD algorithm defines a minimisation problem over $\gamma_i$ coordinates at each iteration where the rest of the coordinates $\Gamma\backslash\gamma_i$ are regarded as fixed. In this way, the convex and differentiable part for each $\gamma_i$ in the $j$th iteration is

$$f_i^j(x) = \frac{1}{2}\|\mathbf{Z}_{\gamma_i}x - r_{-i}^j\|_2^2 + \zeta\frac{1}{2}\ \|x\|_2^2 \tag{8}$$

where $\zeta = (1 - \alpha)\lambda N$ and,

$$r_{-i}^j = y - \left(\sum_{k=0}^{i-1}\mathbf{Z}_{\gamma_k}\theta_{\gamma_k}^j + \sum_{k=i+1}^{m-1}\mathbf{Z}_{\gamma_k}\theta_{\gamma_k}^{j-1}\right) \tag{9}$$

whereas the convex and non-differentiable part for each $\gamma_i$ does not depend on the $j$th iteration,

$$g_i(x) = \zeta_i\ \|x\|_2, \tag{10}$$

where $\zeta_i = \alpha\lambda N\eta_i$. Finally, at each $j$th iteration, we solve

$$\theta_{\gamma_i}^j = \underset{x}{\mathbf{argmin}}\, f_i^j(x) + g_i(x) \tag{11}$$

*Subgradient equations* Since the function to be minimized in (11) is convex the subgradient equations characterise the optimum $x^* := \theta_{\gamma_i}^j$. Denoting the subgradient operator by $\partial$, we obtain:

$$\nabla f_i^j(x^*) + \partial g_i(x^*) = 0. \tag{12}$$

We compute $\nabla f_j^i(x)$ from (8),

$$\nabla f_i^j(x) = \mathbf{Z}_{\gamma_i}^T(\mathbf{Z}_{\gamma_i}x - r_{-i}^j) + \zeta x. \tag{13}$$

Also, from (10) we have that $\partial g_i = \zeta_i\partial\|x\|_2$, where

$$\partial\|x\|_2 = \begin{cases}\dfrac{x}{\|x\|_2} & \text{if}\quad x \neq \mathbf{0}\\ \{u\colon \|u\|_2 \leqslant 1\} & \text{if}\quad x = \mathbf{0}\end{cases} \tag{14}$$

Finally, solving (12) for $x^* = \mathbf{0}$, we obtain a discard condition that avoids computing the optimization problem in (11) when the solution is zero:

$$\|\mathbf{Z}_{\gamma_i}^T r_{-i}^j\|_2 \leqslant \zeta_i. \tag{15}$$

If (15) is not satisfied, an accelerated proximal gradient method is performed; otherwise, $\theta_{\gamma_i}^j = \mathbf{0}$.

### *2.2.2. Accelerated proximal gradient method (APGM)*

Proximal gradient methods are useful for optimising convex composite functions, that is, functions which are the sum of a differentiable term and a non-differentiable term, e.g. (11). To implement it, the proximal operator of the non-differentiable term is applied to the update step of the gradient descent algorithm of the differentiable term. In our case, the updating rule at the $k$th step is:

$$y^k = \underset{t_i g_i}{\mathbf{prox}}(y^{k-1} - t_i\ \nabla f_i^j(y^{k-1})), \tag{16}$$

where $t_i \in (0,\ \ 1/L_i]$ ensures convergence of $\|y^k - \theta_{\gamma_i}^j\| \to \mathbf{0}$ being $L_i$ the Lipchitz constant of $\nabla f_i^j$, and the proximal operator of the non-differentiable term is

$$\underset{t_i g_i}{\mathbf{prox}}(z) = \left(1 - t_i\frac{\zeta_i}{\|z\|_2}\right)_+ z, \tag{17}$$

where $(\cdot)_+ := \mathbf{max}(\cdot{,}0)$.

*Lipchitz constant* The function $f_i^j$ is smooth, that is, it has a Lipschitz continuous gradient which ensures the existence of $L_i$. To compute it, we can use the additivity property of the Lipchitz operator applied to (13):

$$L_i = \|\mathbf{Z}_{\gamma_i}\|_2^2 + \zeta, \tag{18}$$

where $\|X\|_2$ is the matrix norm induced by the $\ell_2$ norm.

*Nesterov's acceleration* A Nesterov momentum is performed to reduce the iterations. Equation (16) evaluated on $y^{k-1}$ preceed this rule:

$$x^k = y^k + \frac{k}{k+3}(y^k - y^{k-1}). \tag{19}$$

## 2.3. Implementation details

In this subsection, we explain the algebraic operations involved in both loops of the algorithm and the convergence criteria used in each case. In addition, we show the simplifications implemented to reduce the use of memory and calculation time in matrix-vector and matrix-adjoint-vector operations. Also, we present details of the range of hyperparameters, initialisation strategies, appropriate convergence rates, and information related to algorithm programming.

### *2.3.1. External loop*

At the external loop (BCD iteration) we need to check the null condition (15) for each group $\gamma_i$, and the stopping rule after a full cycle.

We do not compute $r_{-i}^j$ directly from (9); instead, we approach its value recursively. We set a variable $r_{ext}$ which has the role of partial residue (9) before evaluating the null condition (15), and the role of total residue afterwards.

Starting from the total residue $r_{ext} = y - \mathbf{vec}(\mathbf{A}\boldsymbol{\Theta}^0\mathbf{D}^T)$, we compute the partial residue $r_{-i}^j$ as:

$$r_{ext} \leftarrow r_{ext} + \mathbf{Z}_{\gamma_i}\theta_{\gamma_i}^{j-1}. \tag{20}$$

After evaluating the null condition and the optimisation step, we obtain $\theta_{\gamma_i}^j$ and turn $r_{ext}$ into the total residue by,

$$r_{ext} \leftarrow r_{ext} - \mathbf{Z}_{\gamma_i}\theta^j_{\gamma_i}. \tag{21}$$

The aforementioned process allows us to compute $r^j_{-i}$ and avoids performing the product $\mathbf{Z}\theta^j$. The BCD algorithm is shown in Alg. 1.

**Algorithm 1.** Block coordinate descent

```
Require: $\mathbf{Z} \in \mathbf{R}^{pn\times qk}$, $\theta^0 \in \mathbf{R}^{qk}$, $y \in \mathbf{R}^{pn}$, $\alpha \in [0,\ 1]$,
    $\lambda > 0$, $\beta \in (0,\ 1]$, $(\gamma_i,\ \eta_i)$   for $i = 0,\dots, m - 1$
    $N \leftarrow pn$
    $\zeta \leftarrow (1 - \alpha)\lambda N$
    $\zeta_i \leftarrow \alpha\lambda N\eta_i \quad \forall i$
    $L_i \leftarrow \|\mathbf{Z}_{\gamma_i}\|_2^2 + \zeta \quad \forall i$      ▷Lipchitz constants
    $t_i \leftarrow \beta/L_i \quad \forall i$      ▷step sizes
    $j \leftarrow 1$
    $r_{ext} \leftarrow y - \mathbf{Z}\theta^0$
    repeat
        for $i \leftarrow 0$ to $m$ do      ▷external loop
            $r_{ext} \leftarrow r_{ext} + \mathbf{Z}_{\gamma_i}\theta^{j-1}_{\gamma_i}$,      ▷partial residual
            if$\|\mathbf{Z}^T_{\gamma_i}r_{ext}\|_2 \leqslant \zeta_i$then      ▷null condition
                $\theta^j_{\gamma_i} \leftarrow$ APGM $\mathbf{Z}_{\gamma_i}, \theta^{j-1}_{\gamma_i}, r_{ext}, \zeta_i, \zeta, t_i$
                $r_{ext} \leftarrow r_{ext} - \mathbf{Z}_{\gamma_i}\theta^j_{\gamma_i}$      ▷total residual
            else
                $\theta^j_{\gamma_i} \leftarrow \mathbf{0}$
            endif
        end for
        $j \leftarrow j + 1$
until $\max|\Delta r_{ext}| < \epsilon_{ext}$      ▷stopping criteria
```

*Stopping criterion* We consider that the algorithm has converged when the maximum change in the total residue between two full cycles is less than a predefined tolerance, $\epsilon_{ext}$,

$$\max|\Delta r_{ext}| < \epsilon_{ext}. \tag{22}$$

### 2.3.2. Internal loop

At the internal loop involved in the APGM, we compute the total residue $r_{int}$ recursively to evaluate the gradient (13). Initialising at $r_{int} \leftarrow r^j_{-i} - \mathbf{Z}_{\gamma_i}\theta^0_{\gamma_i}$,

$$r_{int} \leftarrow r_{int} - \mathbf{Z}_{\gamma_i}(x^k - x^{k-1}), \tag{23}$$

and we compute (13) as,

$$\nabla f_i^k = -\mathbf{Z}^T_{\gamma_i}r_{int} + \zeta x^k. \tag{24}$$

The APGM algorithm is shown in Alg. 2.

**Algorithm 2.** Accelerated proximal gradient method

```
Require: $\mathbf{Z} \in \mathbf{R}^{n\times m}$, $x^0 \in \mathbf{R}^m$, $y \in \mathbf{R}^n$, $\alpha, \beta, \gamma$
    $k \leftarrow 1$
    $y^0 \leftarrow x^0$
    $r_{int} \leftarrow y - \mathbf{Z}x^0$
    repeat      ▷internal loop
        $\nabla f^{k-1} \leftarrow -\mathbf{Z}^T r_{int} + \beta y^{k-1}$
        $z \leftarrow y^{k-1} - \gamma\ \nabla f^{k-1}$
        $y^k \leftarrow \left(1 - \gamma\frac{\alpha}{\| z \|_2}\right)_+ z$      ▷proximal operator
```

```
(Continued.)
        $x^k \leftarrow y^k + \frac{k}{k+3}(y^k - y^{k-1})$      ▷Nesterov's
    acceleration
        $r_{int} \leftarrow r_{int} - \mathbf{Z}\Delta x$ ▷update residual
        $y^k \leftarrow x^k$
        $k \leftarrow k + 1$
    until $\left\| \frac{\Delta x}{\gamma} \right\|_2 < \epsilon_{int}$      ▷stopping criteria
```

*Stopping criterion* The inner loop is stopped when the norm of the generalised gradient is less than a predefined tolerance $\epsilon_{int}$,

$$\left\| \frac{\Delta x}{t_i} \right\|_2 < \epsilon_{int}, \tag{25}$$

where $t_i$ is the step size of the i-group.

### 2.3.3. Matrix-vector products

In this section, we focus on the structure of $\mathbf{Z}_{\gamma_i}$, with the aim of reducing the time and memory burden involved in the computation of the products $\mathbf{Z}_{\gamma_i}u$ and $\mathbf{Z}^T_{\gamma_i}v$. The key is to avoid computing $\mathbf{Z}_{\gamma_i}$ explicitly. To achieve this, we return to (3) where $\mathbf{Z} = \mathbf{D} \otimes \mathbf{A}$, and we note that its $r$-column is given by $Z_r = D_{\lfloor r/q \rfloor} \otimes A_{r\%q}$, which is a column-wise Kronecker product. This leads to the Khatri-Rao product definition for the group $\gamma_i \in \Gamma$,

$$\mathbf{Z}_{\gamma_i} = \mathbf{D}_{\lfloor \gamma_i/q \rfloor}{}^*\mathbf{A}_{\gamma_i\%q}, \tag{26}$$

the division /, module % and floor $\lfloor\ \rfloor$ operators are entry-wise applied over $\gamma_i$. The Khatri-Rao product property $(\mathbf{B}^*\mathbf{A})u = \mathbf{vec}(\mathbf{A}\mathbf{U}_d\mathbf{B}^T)$ where $\mathbf{U}_d$ is a diagonal matrix, allows us to obtain $\mathbf{Z}_{\gamma_i}u$ without computing the product of $\mathbf{Z}_{\gamma_i}$ and $u$. However, we cannot use the same property to avoid the computation $\mathbf{Z}^T_{\gamma_i}v$. To overcome this issue, we set a constraint to the set of partitions of $\Phi$.

In figure 1 we show three examples of partitions $\Gamma$ of the set $\{0,\dots,19\}$. Each partition contains six groups and are represented in a matrix. Each group is a set of numbers which represents a set of entries of $\theta$. To facilitate the visualisation, each group is identified with a texture. In panel **A**, we observe that an unstructured partition implies that the computation of $\mathbf{Z}_{\gamma_i}$ requires the Kathri-Rao product.

In panel **B**, we choose a partition that verifies $\lfloor \gamma_i/q \rfloor = \{r_i,\dots,r_i\}$ for all $\gamma_i \in \Gamma$. Graphically, we observed that each group/texture is contained in a single column. Under this assumption, $\mathbf{D}_{\lfloor \gamma_i/q \rfloor}$ has repeated columns $D_{r_i}$ and the Khatri-Rao product in (26) becomes the Kronecker product:

$$\mathbf{Z}_{\gamma_i} = D_{r_i} \otimes \mathbf{A}_{\gamma_i\%q}. \tag{27}$$

On one hand, from the Kronecker product property, we obtain $\mathbf{Z}_{\gamma_i}u = \mathbf{vec}(\mathbf{A}_{\gamma_i\%q}\mathbf{U}D^T_{r_i})$, but in this case $\mathbf{U} = u$, so a matrix-vector product and an outer product were performed. However, we compute

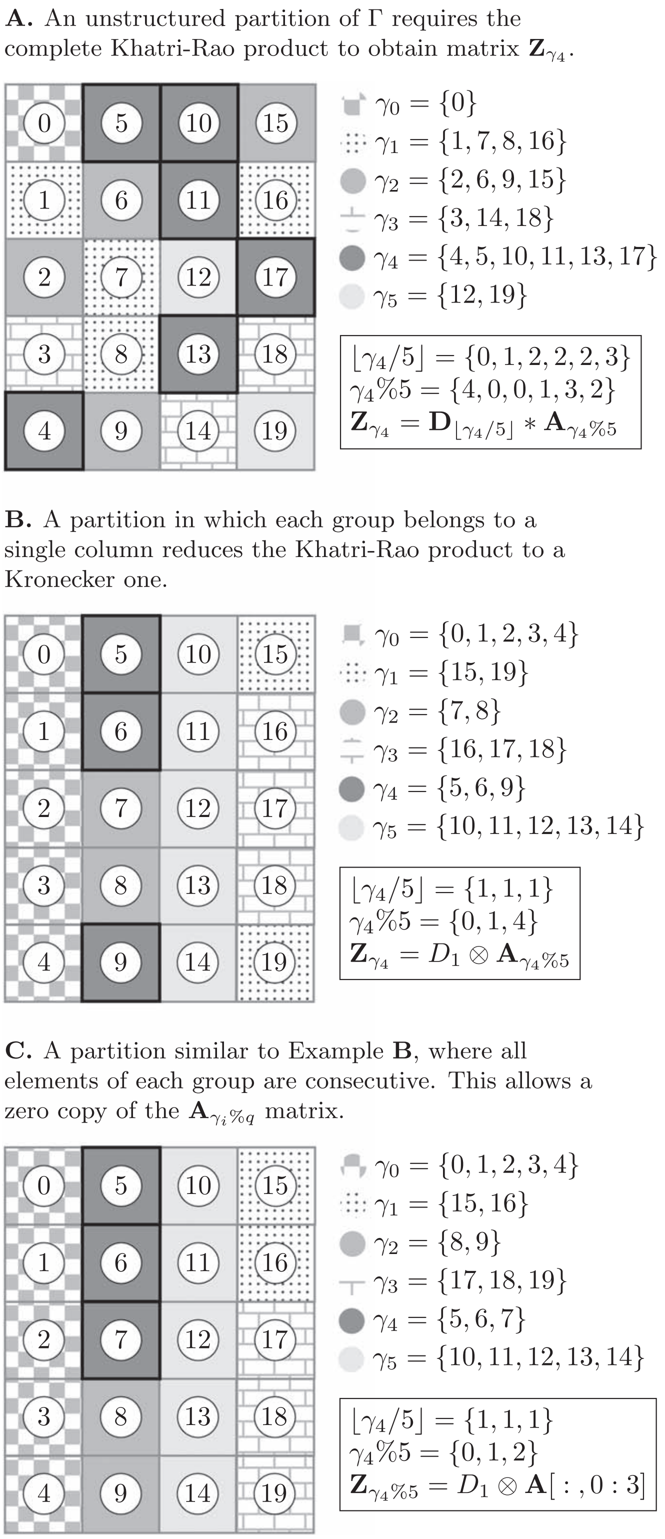
**A.** An unstructured partition of Γ requires the complete Khatri-Rao product to obtain matrix $\mathbf{Z}_{\gamma_4}$.
0 5 10 15
1 6 11 16
2 7 12 17
3 8 13 18
4 9 14 19
$\gamma_0 = \{0\}$
$\gamma_1 = \{1, 7, 8, 16\}$
$\gamma_2 = \{2, 6, 9, 15\}$
$\gamma_3 = \{3, 14, 18\}$
$\gamma_4 = \{4, 5, 10, 11, 13, 17\}$
$\gamma_5 = \{12, 19\}$
$\lfloor \gamma_4/5 \rfloor = \{0, 1, 2, 2, 2, 3\}$
$\gamma_4 \% 5 = \{4, 0, 0, 1, 3, 2\}$
$\mathbf{Z}_{\gamma_4} = \mathbf{D}_{\lfloor \gamma_4/5 \rfloor} * \mathbf{A}_{\gamma_4 \% 5}$
**B.** A partition in which each group belongs to a single column reduces the Khatri-Rao product to a Kronecker one.
0 5 10 15
1 6 11 16
2 7 12 17
3 8 13 18
4 9 14 19
$\gamma_0 = \{0, 1, 2, 3, 4\}$
$\gamma_1 = \{15, 19\}$
$\gamma_2 = \{7, 8\}$
$\gamma_3 = \{16, 17, 18\}$
$\gamma_4 = \{5, 6, 9\}$
$\gamma_5 = \{10, 11, 12, 13, 14\}$
$\lfloor \gamma_4/5 \rfloor = \{1, 1, 1\}$
$\gamma_4 \% 5 = \{0, 1, 4\}$
$\mathbf{Z}_{\gamma_4} = D_1 \otimes \mathbf{A}_{\gamma_4 \% 5}$
**C.** A partition similar to Example **B**, where all elements of each group are consecutive. This allows a zero copy of the $\mathbf{A}_{\gamma_i \% q}$ matrix.
0 5 10 15
1 6 11 16
2 7 12 17
3 8 13 18
4 9 14 19
$\gamma_0 = \{0, 1, 2, 3, 4\}$
$\gamma_1 = \{15, 16\}$
$\gamma_2 = \{8, 9\}$
$\gamma_3 = \{17, 18, 19\}$
$\gamma_4 = \{5, 6, 7\}$
$\gamma_5 = \{10, 11, 12, 13, 14\}$
$\lfloor \gamma_4/5 \rfloor = \{1, 1, 1\}$
$\gamma_4 \% 5 = \{0, 1, 2\}$
$\mathbf{Z}_{\gamma_4 \% 5} = D_1 \otimes \mathbf{A}[\,:\,, 0 : 3]$

**Figure 1.** Three partitions Γ showing the relation between the group structure and the impact on the computation of $\mathbf{Z}_{\gamma_4}$. Each $\gamma_i$ is a set of numbers which represents a set of entries of $\theta$ and are identified with same texture. The black border highlights the group number four and the associated modulus and division operations. (a) An unstructured partition of Γ requires the complete Khatri-Rao product to obtain matrix $\mathbf{Z}_{\gamma_4}$. (b) A partition in which each group belongs to a single column reduces the Khatri-Rao product to a Kronecker one. (c) A partition similar to Example **B**, where all elements of each group are consecutive. This allows a zero copy of the $\mathbf{A}_{\gamma_i \% q}$ matrix.

$\mathbf{Z}_{\gamma_i}^T v = \mathbf{vec}(\mathbf{A}_{\gamma_i \% q}^T \mathbf{V} D_{r_i})$ performing two matrix-vector products.

Finally, in panel **C**, we add the hypothesis that the groups have contiguous numbers (this is visualised as non-interrupted textures), which allows us to make a zero-copy evaluation of each $\mathbf{A}_{\gamma_i \% q}$ if **A** has a Fortran memory order. Mathematically, we consider only the partitions that satisfy $\gamma_i\%q = [a_i, b_i]$ for all $\gamma_i \in \Gamma$ denoting $[a, b] := \{a, a+1,...,b-2, b-1\}$.

*2.3.4. Hyperparameters*

Commonly, linear regression problems are performed on a grid of hyperparameters and some heuristic criterion is defined to pick a unique solution $(\alpha^*, \lambda^*)$. In this work, we set a fixed $\alpha^*$ and the range of $\lambda$'s is determined by the ratio $\epsilon = \lambda_{\min}/\lambda_{\max}$. If the null condition (15) of each group is evaluated at $\mathbf{0}$, we obtain the value of $\lambda$ to deactivate the group. Therefore, by taking the maximum of these $\lambda$ values, we obtain a null solution. Mathematically, we compute this value as

$$\lambda_{\max} = \frac{1}{N\alpha^*} \max_i \frac{\|\mathbf{Z}_{\gamma_i}^T y\|_2}{\eta_i} \tag{28}$$

Finally, the values of $\lambda$ are organized in decreasing order on a logarithmic scale. For $\alpha^* = 0$, we used a manual range.

*Warm start* To accelerate the convergence of the algorithm, we initialise the $(\alpha^*, \lambda_{i+1})$-optimisation problem with the optimum solution of the previous target $(\alpha^*, \lambda_i)$. Note that the values of $\lambda$ are in the decreasing order and $\lambda_{\max}$ promotes a null solution.

*Step size* $(t_i)$ We set a fixed step size $t_i = \beta/L_i$ with $\beta \in (0, 1]$ for each group $\gamma_i$ in the proximal iteration. We use the property $\|\mathbf{X}_1 \otimes \mathbf{X}_2\|_2 = \|\mathbf{X}_1\|_2\|\mathbf{X}_2\|_2$, and (27) to compute (18).

*Weight* $(\eta_i)$ The weight $\eta_i$ is computed as $\sqrt{p_i}$ being $p_i = \mathbf{card}(\gamma_i)$ [20].

*2.3.5. Programming*

Algorithms 1 and 2 were coded in Cython language [28]. We used the Basic Linear Algebra Subprograms (BLAS) [29] routines through SciPy [30] wrappers in all algebraic operations, which drastically reduces the processing time compared to the NumPy implementation [31].

**2.4. Experiments**

We used two different kinds of data. Firstly, random data allowed us to validate the proper functioning of the algorithm. Then we extracted data from the EDGAR database, which is used for research and validation of ECGI algorithms [32].

*2.4.1. Algorithm testing*

In this first experiment, we simulated a matrix with the Kronecker structure, $\mathbf{Z} = \mathbf{D} \otimes \mathbf{A}$, with $\mathbf{D}$ and $\mathbf{A}$ random matrices of size $10 \times 10$ with zero-mean and unit-variance normal i.i.d. in each element.

To obtain simulated solutions $\Theta_s \in {}'R^{10\times 10}$, we created a dense matrix with normal distribution in each element and we applied a mask with one group per column and three levels of group sparsity: 90 %, 50 % and 0 %. The right panel of figure 1 shows the simulated data with the coefficient values in grey scale.

We established values for $\alpha$: $\alpha = 1$ (group *lasso*), $\alpha = 0.5$ and $\alpha = 0$ (ridge regression), and we computed the measurements as $y_s = \mathbf{Z}\theta_s$ for each solution. Then, we defined a hundred points for the range of $\lambda$ and set $\epsilon = 10^{-4}$. If $\alpha = 0$, the range is $[10^{-5}, 10^0]$. Finally, the convergence rates are scaled with $\beta = 0.8$.

From the above configuration, we executed nine optimisations composed of three different values of $\alpha$ for each group sparsity. Section 3 presents the norm of the coefficients for each group and the relative error to analyse the output of the algorithm. We define the error as:

$$\text{error} = \frac{\|\Theta - \Theta_s\|_1}{\|\Theta_s\|_1} \tag{29}$$

*2.4.2. ECGI application*

We compared the proposed regularization against zero-order Tikhonov (Tkh0) regularization based on several metrics. To achieve this, we selected the following EDGAR datasets: (1) *Maastricht-09-15-06*: It contains a sinus beat and another stimulated in the apex of the left ventricle of an anaesthetized dog, sampled at 2 kHz. Sinus beat contains 140 ECGs with one interpolated (to poor signal quality) and 83 epicardium noise-free channels. At the same time, there are 135 electrodes located on the torso and 65 epicardium noise-free electrodes for the ELG recording of the stimulated heartbeat. The heart mesh contains 1321 nodes. We refer to this dataset as sinus dog (SD) and paced dog (PD). (2) *Auckland-2012-06-05*: It contains multiple sinus and paced beats at the epicardium of a pig. We used the first beat of both procedures, sampled at 2 kHz. The sinus beat experiment contains 171 electrodes on the torso, 30 of them were interpolated (to poor signal quality) and 224 epicardium noise-free channels. At the same time, there are 171 electrodes located on the torso with 29 ECGs that were interpolated and 226 epicardium noise-free channels. The heart mesh contains 1502 nodes. We refer to this dataset as sinus pig (SP) and paced pig (PP). (3) *KIT-20-PVC Simulation-1906-10-30 EP Peri*: It is a dataset with ECGs and extracellular potentials simulated computationally with FEM and the bidomain model. It contains a stimulated heartbeat in different areas of the heart of a human. For this study, we have used the stimulated beat in the lateral endocardial zone of the left ventricle. It contains 163 electrodes on the torso and 502 electrodes/nodes on the pericardium. We refer to this dataset as paced human (PH).

For all datasets, we created the torso mesh using the electrodes as nodes, and the faces have been built with the provided ball pivoting algorithm by MeshLab generating a closed surface [33]. The datasets provide the mesh of the heart.

Once the geometries were defined, we computed the matrix $\mathbf{A}$ solving a Laplace problem with Cauchy boundary conditions [8] using Bempp [34].

We interpolated torso channels with low SNR using the Laplace method [35] whereas we discarded cardiac channels under the same conditions.

To perform the experiments, we used two dictionaries. First, we applied a wavelet dictionary *db2* (Daubechies with two vanishing moments). It contains six levels of detail and a scaling function for the approximation level with a step between atoms of 1% of the temporal samples of the signal. Secondly, we built a physiological dictionary (Phy) from the real electrograms in order to test a more customized frame.

We implemented $\alpha = 0.95$, a hundred values of $\lambda$, $\epsilon = 10^{-5}$ and $\beta = 0.8$. We selected $\lambda^*$ such that the median of the spatial correlation coefficient is maximised, and each group contains all the cardiac nodes. To compare our results with the gold standard, we applied Tikhonov regularisation of order 0 using a hundred values of $\lambda$ with $\lambda_{\max} = 10\|\mathbf{A}\|_2$ and $\lambda_{\min} = 0.1\|\mathbf{A}\|_{-2}$ in decreasing log-space order.

We compared the ELG signals estimated from the inverse problem solution with the ELG signals measured on the heart surface. Whenever the electrode position did not coincide with the node position, we selected the node with the smallest Euclidean distance to the electrode.

We used temporal and spatial cross-correlation as an ELGs morphologic metric, and we computed the spatial and temporal relative error to quantify the differences in the amplitudes of the ELGs. Also, we calculated the absolute error of the activation times, estimated as the sample where the smallest ELG derivative occurs around the QRS complex.

## 3. Results

Based on the experiment defined in section 2.4.1, figure 2 shows the behaviour of the solutions versus the values of $\lambda$ for a grid of values of group sparsity and $\alpha$.

From this same experiment, we obtained the information to compute the last row of 2, which shows the performance of the error of estimation of the solutions for the same grid of values.

When the group sparsity was 90% (see the first column in figure 2), group 3 showed predominance, increasing the norm value, while values of $\lambda$ decreased. We observed this tendency for three different values of $\alpha$. Similar results were observed when the group sparsity was 50% and 0% (second and third columns in figure 2) with its corresponding groups 0, 2, 3, 4 and 6.

The last row of figure 2 shows the behaviour of the error computed as a function of $\lambda$, using (29). It is expected that for group sparsity values greater than 50% the solutions of $\alpha = 0$ will yield a larger error than those with the sparse constraint. Conversely, when the group sparsity is less than 50% we observed that the error decreased when $\alpha = 0$. In the critical case, group sparsity of 50%, the magnitude of the error depended on the value of $\lambda$, as we can observe in the middle panel of the last row of figure 2.

Comparing the error curves for the cases $\alpha = 0.5$ and $\alpha = 1.0$, we observed that the error for $\alpha = 1.0$ was always larger than the error corresponding to $\alpha = 0.5$, regardless of the group sparsity. However, both curves differed only by the $\lambda$ offset. We also observed that, given a group sparsity, the curves of the norm of the coefficients for $\alpha = 1.0$ and $\alpha = 0.5$ were very similar.

The results obtained from the experiment described in section 2.4.2 are showed in figure 3(a) and table 1.

In figure 3(a), we show the morphological comparison of four electrograms recovered with Tikhonov of order 0 and the algorithm presented in this work with $\alpha = 0.95$ for Daubechies and for an implementation based on physiological information. The physiological dictionary was built using real electrograms. It showed better high frequency filtering compared to Tikhonov and Daubechies. Table 1 presents the 1st, 2nd and 3rd quartiles of each one of the five metrics for each experiment. These results are consistent, according to the morphological similarities observed in figure 3(a). In general, the computed ELGs have the half amplitude of the ground truth, and relative error near to one. The cross-correlation values and activation time error depend on the experiment, but they remain close to the gold standard.

In addition to the memory and time savings produced by avoiding the explicit computation of $\mathbf{Z}_{\gamma_i}$, we observed that the operation complexity to compute $\mathbf{Z}_{\gamma_i}u$ and $\mathbf{Z}_{\gamma_i}^T v$ was reduced from $\mathcal{O}(pp_i n)$ to $\mathcal{O}(p(p_i + n))$. For instance, in the case of the dog sinus beat where $p = 140$ (torso electrodes), $p_i = 1321$ (heart nodes of each group) and $n = 593$ (cardiac beat length), a 400-fold reduction in required operations can be calculated for both products.

## 4. Discussion

In this work, we present an algorithm with the capacity to use redundant dictionaries, as well as the possibility of using different groups of nodes.

The numerical implementation of an algorithm that operates at acceptable times has been challenging. To do this we took advantage of the Kronecker structure of the matrix $\mathbf{Z}$, reducing the memory required to compute the model. In addition, by applying restrictions on the distribution of groups, we were able to reduce the complexity of the algebraic operations. However, despite the applied optimizations, a low-level implementation was necessary to reduce the computation times in the algebraic operations.

Tikhonov regularization has several mechanisms for selecting its optimal hyperparameters such as L-Curve, CRESO and U-Curve [36]. However, as it was claimed in [21], in regularizations that include

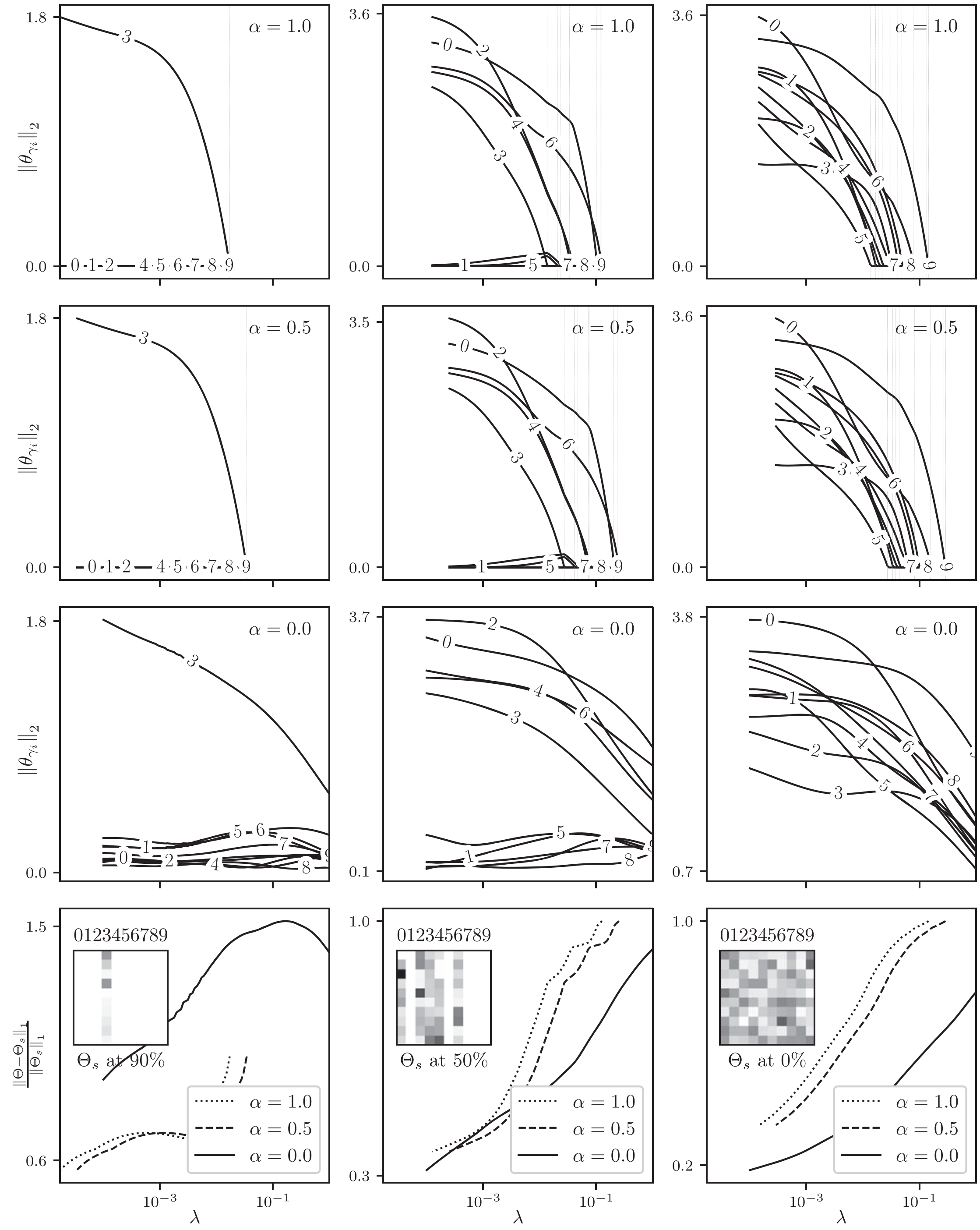

**Figure 2.** Graphics showing the 2-norm of each group of coefficients versus $\lambda$ and the relative errors versus $\lambda$. Each curve is associated with a number indicating the corresponding group. The matrices, with their corresponding groups, are presented in the last row of this figure. The vertical lines indicate the values of $\lambda$ for which a group has a non-zero norm. The rows are parametrized by different values of $\alpha$ (1.0, 0.5 and 0.0) and the columns by the level of group sparsity (90%, 50% and 0%). The ground truth coefficients are displayed on the last row along with the relative error curves as a function of $\lambda$. All graphs share the same abscissa axis.

more than one hyperparameter there are no standard criteria for selecting them.

In the present work, we have fixed the hyperparameter $\alpha = 0.95$ according to the following observations. On the one hand, the results show that for $\alpha \neq 0$ errors have a similar behaviour taking into account an offset in $\lambda$, as is shown in the last row of figure 2. We have taken $\alpha \approx 1$ because the value of $\lambda_{\max}$ increases as $\alpha$ goes to zero (see 28), generating a large number of over smoothness and low amplitude by the predominance of the $\ell_2$ term. On the other hand, we have empirically observed a higher speed of convergence when $\alpha \neq 1$.

Likewise, the choice of $\lambda^*$ is based on the maximization of the spatial cross-correlation. We highlighted that this procedure cannot be used in applications due to the lack of ground truth. The criteria used for the selection of hyperparameters limit

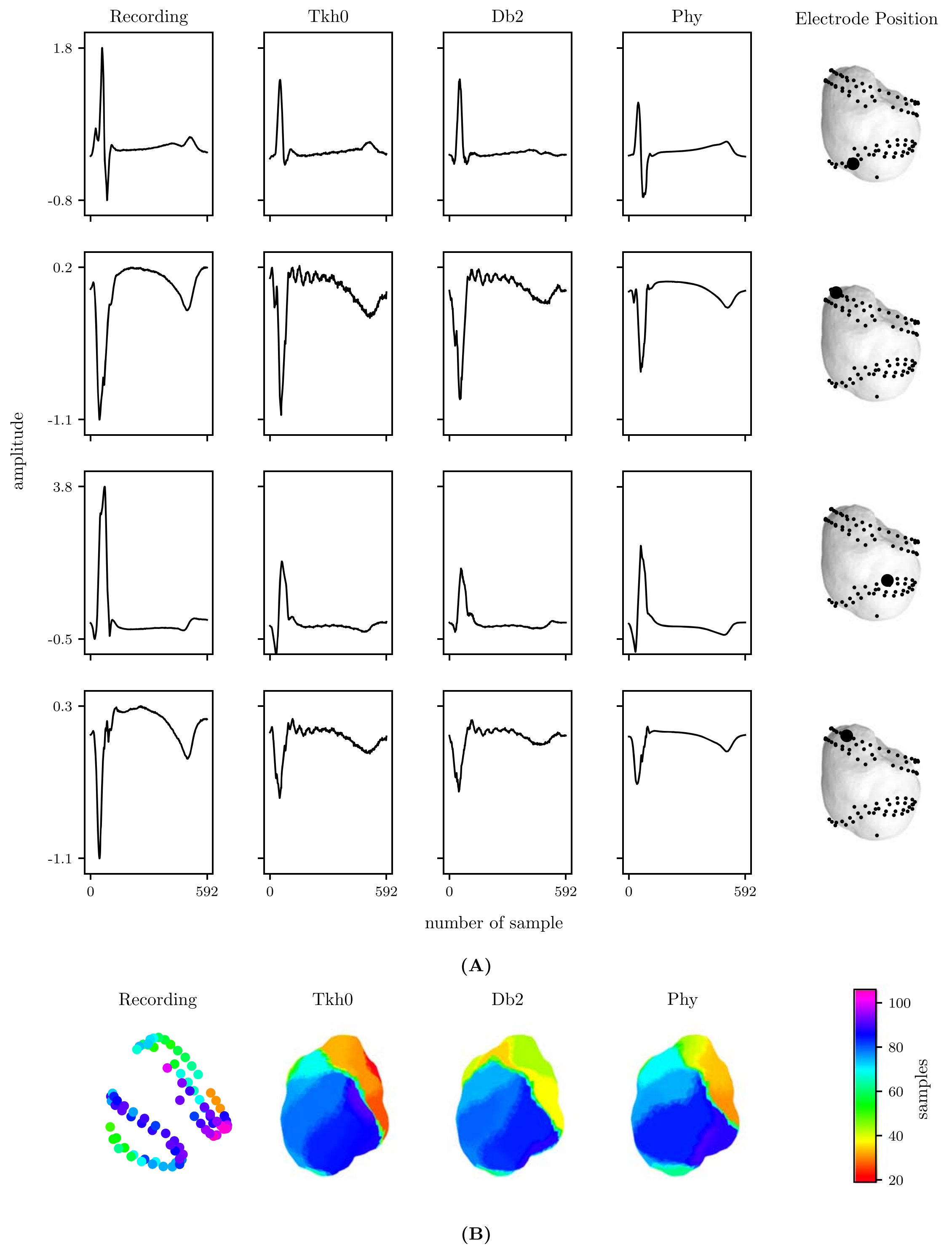

**Figure 3.** Graphics showing solutions to the experiment of dog sinus beat using the three optimizations under consideration: Tikhonov (Tkh0), Daubechies with two vanishing moments (Db2), and a physiological dictionary obtained from real electrograms (Phy). Panel (**A**) shows four ELGs, one in each row. The first column shows the morphology of the selected ELG (Recording). The last column presents the spatial location (Electrode Position). Panel (**B**) shows the spatial visualization of the activation times (i.e. propagation maps) for the experiment of the dog sinus beat together with its corresponding ground-truth (Recording).

the comparison of the regularisations. In this sense, comparing regularisations taking their optimal hyperparameters based on some metric would be a good starting point. But other problems can appear, such as (the value of) the hyperparameter that maximizes the temporal cross-correlation may not coincide with the hyperparameter that maximizes the spatial cross-correlation. Table 1 shows that the metrics are low for all regularizations, including the ECGI gold standard. These are the most important limitations of the technique. In this sense, the development of new regularizations is imperative.

The proposed regularization presents the possibility of using dictionaries built from features with

**Table 1.** Metrics comparison for all experiments. The first column indicates the type of experiment: sinus dog (*SD*), paced dog (*PD*), sinus pig (*SP*) paced pig (*PP*) and paced human (*PH* ). The second column indicates the corresponding metric: cross-correlation (*CC*), relative error (*RE*) and absolute error of the activation times in samples (*EAT*); subscript *t* stands for temporal and subscript *s* stands for spatial. Results are shown as Q2(Q1-Q3), where Qi is the *i-th* quartile.

| | | **Tkh0** | **Db2** | **Phy** |
|---|---|---|---|---|
| **SD** | $CC_t$ | 0.76(0.61–0.85) | 0.76(0.63–0.86) | 0.76(0.51–0.89) |
| | $CC_s$ | 0.43(0.37–0.50) | 0.43(0.35–0.49) | 0.44(0.39–0.50) |
| | $RE_t$ | 0.86(0.71–0.96) | 0.86(0.75–0.97) | 0.87(0.73–0.97) |
| | $RE_s$ | 0.90(0.89–0.95) | 0.91(0.90–0.97) | 0.89(0.88–0.93) |
| | $EAT$ | 11.0(6.0–22.0) | 11.0(7.5–19.5) | 10.0(4.5–16.0) |
| **PD** | $CC_t$ | 0.71(0.57–0.82) | 0.73(0.52–0.83) | 0.78(0.58–0.88) |
| | $CC_s$ | 0.56(0.45–0.62) | 0.56(0.43–0.63) | 0.63(0.50–0.66) |
| | $RE_t$ | 0.89(0.77–0.99) | 0.91(0.84–0.98) | 0.89(0.79–0.97) |
| | $RE_s$ | 0.88(0.87–0.92) | 0.90(0.86–0.97) | 0.88(0.87–0.92) |
| | $EAT$ | 10.5(4.0–15.2) | 7.0(2.8–13.2) | 7.0(2.0–12.2) |
| **SP** | $CC_t$ | 0.85(0.63–0.96) | 0.78(0.47–0.89) | 0.85(0.40–0.96) |
| | $CC_s$ | 0.38(0.29–0.55) | 0.39(0.29–0.60) | 0.39(0.24–0.52) |
| | $RE_t$ | 0.91(0.85–0.97) | 0.98(0.95–1.00) | 0.99(0.97–1.00) |
| | $RE_s$ | 0.95(0.92–0.96) | 1.00(1.00–1.00) | 1.00(0.98–1.00) |
| | $EAT$ | 16.0(7.8–29.0) | 15.0(6.0–30.0) | 16.0(6.0–30.0) |
| **PP** | $CC_t$ | 0.95(0.85–0.98) | 0.92(0.81–0.97) | 0.94(0.84–0.98) |
| | $CC_s$ | 0.61(0.41–0.74) | 0.63(0.36–0.72) | 0.64(0.46–0.73) |
| | $RE_t$ | 0.72(0.53–0.86) | 0.70(0.55–0.89) | 0.70(0.55–0.86) |
| | $RE_s$ | 0.85(0.73–0.95) | 0.87(0.73–0.98) | 0.83(0.74–0.94) |
| | $EAT$ | 11.0(5.0–23.0) | 19.0(8.0–28.8) | 12.0(5.0–22.0) |
| **PH** | $CC_t$ | 0.64(0.48–0.83) | 0.64(0.49–0.82) | 0.62(0.45–0.82) |
| | $CC_s$ | 0.38(0.27–0.44) | 0.29(0.16–0.43) | 0.29(0.20–0.41) |
| | $RE_t$ | 0.92(0.68–0.98) | 0.93(0.70–0.98) | 0.93(0.70–0.98) |
| | $RE_s$ | 0.95(0.91–0.97) | 0.98(0.93–1.00) | 0.96(0.93–0.98) |
| | $EAT$ | 29.0(9.0–52.8) | 31.0(12.0–49.0) | 29.0(11.0–48.8) |

physiological meaning. For example, we may obtain dictionaries trained with physiologic data using supervised machine learning algorithms, or more complex wavelets dictionaries without orthogonality restrictions. Finally, the algorithm presented in this work can be also used in other applications where the matrix in the model has a Kronecker structure.

This work aims to extend two concepts based on [21]: (1) the advantage of using a more versatile group distribution than a single group per column, for example, defining groups with nearby nodes. (2) The possibility of using any type of dictionary, without the restriction of a tight frame. This may allow to add physiological information into the regularisation.

## 5. Conclusion

We efficiently implemented an optimisation algorithm to solve the group *lasso* problem plus the Tikhonov term for models with the Kronecker structure. We developed an algorithm that supports the use of arbitrary dictionaries to obtain solutions and allows flexible group distributions.

## 6. Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this study.

## Acknowledgments

This work was supported by grants from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Universidad de Buenos Aires (UBA), Universidad Nacional de La Plata (UNLP), and Agencia Nacional de Promoción Cientıfica y Tecnológica (MINCyT), all of them from Argentina.

P.D. Arini and S.F. Caracciolo work was supported by grants UBACyT 20 020 130 100 485BA, MINCyT PICT 2145-2016, and CONICET PIP 112-20130100552CO.

C. F. Caiafa was partially supported by grants PICT 2017-3208, PICT 2020-SERIEA-00457, UBACyT 20 020 190 200 305BA and UBACyT 20 020 170 100 192BA.

F.D. Martínez Pería gratefully acknowledges the support from grants UNLP 11X829, MINCyT PICT 1505-2015 and CONICET PIP 112-20200102127CO.

## Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

## ORCID iDs

S F Caracciolo https://orcid.org/0000-0002-6347-7186
C F Caiafa https://orcid.org/0000-0001-5437-6095
F D Martínez Pería https://orcid.org/0000-0003-2181-0507
P D Arini https://orcid.org/0000-0002-5548-7528

## References

[1] Bear L R *et al* 2018 How accurate is inverse electrocardiographic mapping? *Circulation: Arrhythmia and Electrophysiology* **11** e006108

[2] Macfarlane P W, van Oosterom A, Pahlm O, Kligfield P, Janse M and Camm J 2010 *Comprehensive Electrocardiology* (London: Springer)

[3] Geselowitz D B 1964 Dipole theory in electrocardiography *The American Journal of Cardiology* **14** 301–6

[4] Geselowitz D B and Miller W T 1983 A bidomain model for anisotropic cardiac muscle *Ann. Biomed. Eng.* **11** 191–206

[5] Milan Horáček B and Clements J C 1997 The inverse problem of electrocardiography: A solution in terms of single- and double-layer sources on the epicardial surface *Mathematical Biosciences.* **144** 119–54

[6] Erem B, van Dam P M and Brooks D H 2014 Identifying model inaccuracies and solution uncertainties in noninvasive activation-based imaging of cardiac excitation using convex relaxation *IEEE Trans. Med. Imaging* **33** 902–12

[7] Seger M, Fischer G, Modre R, Messnarz B, Hanser F and Tilg B 2005 Lead field computation for the electrocardiographic inverse problem finite elements versus boundary elements *Comput. Methods Programs Biomed.* **77** 241–52

[8] Stenroos M and Haueisen J 2008 Boundary element computations in the forward and inverse problems of electrocardiography: comparison of collocation and galerkin weightings *IEEE Trans. Biomed. Eng.* **55** 2124–33

[9] Yao B, Pei S and Yang H 2016 Mesh resolution impacts the accuracy of inverse and forward ECG problems *2016 XXXVIII Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* pp 4047–50

[10] Hansen P C 2010 *Discrete inverse problems* (*Fundamentals of Algorithms*) (USA: Society for Industrial and Applied Mathematics) (https://doi.org/10.1137/1.9780898718836)

[11] Hansen P C 1990 Truncated singular value decomposition solutions to discrete Ill-posed problems with ill-determined numerical rank *SIAM J. Sci. Stat. Comput.* **11** 503–18

[12] Coll-Font J, Wang L and Brooks D H 2018 A common-ground review of the potential for machine learning approaches in electrocardiographic imaging based on probabilistic graphical models *Computing in Cardiology* 45

[13] Dogrusoz Y S 2019 Statistical estimation applied to electrocardiographic imaging *2019 XII International Conference on Measurement* pp 2–9

[14] Cluitmans M J M *et al* 2017 In vivo validation of electrocardiographic imaging *JACC: Clinical Electrophysiology* **3** 232–42

[15] Tibshirani R 1996 Regression shrinkage and selection via the lasso *Journal of the Royal Statistical Society Series B (Methodological)* **58** 267–88

[16] Ghosh S and Rudy Y 2009 Application of l1-norm regularization to epicardial potential solution of the inverse electrocardiography problem *Ann. Biomed. Eng.* **37** 902–12

[17] Shou G, Xia L, Liu F, Jiang M and Crozier S 2010 On epicardial potential reconstruction using regularization schemes with the L1-norm data term *Phys. Med. Biol.* **56** 57–72

[18] Zou H and Hastie T 2005 Regularization and variable selection via the elastic net *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67** 301–20

[19] Yuan M and Lin Y 2006 Model selection and estimation in regression with grouped variables *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68** 49–67

[20] Simon N, Friedman J, Hastie T and Tibshirani R 2013 A Sparse-Group Lasso *Journal of Computational and Graphical Statistics* **22** 231–45

[21] Cluitmans M, Karel J, Bonizzi P, Volders P, Westra R and Peeters R 2018 Wavelet-promoted sparsity for non-invasive reconstruction of electrical activity of the heart *Med. Biol. Eng. Comput.* **56** 2039–50

[22] Cluitmans M J, de Jong M M, Volders P G, Peeters R L and Westra R L 2014 Physiology-based regularization improves noninvasive reconstruction and localization of cardiac electrical activity *Computing in Cardiology* **2014** 1–4

[23] Cluitmans M J M, Clerx M, Vandersickel N, Peeters R L M, Volders P G A and Westra R L 2017 Physiology-based regularization of the electrocardiographic inverse problem *Med. Biol. Eng. Comput.* **55** 1353–65

[24] Cluitmans M J M, Karel J M H, Bonizzi P, Volders P G A, Westra R L and Peeters R L M 2013 Wavelet-sparsity based regularization over time in the inverse problem of electrocardiography *2013 XXXV Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* pp 3781–4

[25] Kovačević J and Chebira A 2008 An introduction to frames *Foundations and Trends® in Signal Processing* **2** 1–94

[26] Tseng P 2001 Convergence of a block coordinate descent method for nondifferentiable minimization *J. Optim. Theory Appl.* **109** 475–94

[27] Hastie T, Tibshirani R and Wainwright M 2015 *Statistical Learning with Sparsity: The Lasso and Generalizations* I edn (Boca Raton: Chapman and Hall/CRC)

[28] Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn D S and Smith K 2011 Cython: the best of both worlds. computing in science engineering *Computing in Science Engineering* **13** 31–9 Conference Name: Computing in Science Engineering

[29] Blackford L S, Pozo R and Al E 2002 An updated set of basic linear algebra subprograms (BLAS) *ACM Trans. Math. Softw.* **28** 135–51 https://www.nist.gov/publications/updated-set-basic-linear-algebra-subprograms-blas

[30] Virtanen P *et al* 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python *Nat. Methods* **17** 261–72

[31] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357–62

[32] Aras K *et al* 2015 Experimental data and geometric analysis repository - edgar *Journal of Electrocardiology* **48** 975–81

[33] Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F and Ranzuglia G 2008 MeshLab: an Open-Source Mesh Processing Tool *Eurographics Italian Chapter Conference The Eurographics Association* 129–36

[34] Betcke T and Scroggs M W 2021 Bempp-cl: A fast Python based just-in-time compiling boundary element library *Journal of Open Source Software* **6** 2879

[35] Oostendorp T F, van Oosterom A and Huiskamp G 1989 Interpolation on a triangulated 3D surface *J. Comput. Phys.* **80** 331–43

[36] Chamorro-Servent J, Dubois R and Coudière Y 2019 Considering new regularization parameter-choice techniques for the tikhonov method to improve the accuracy of electrocardiographic imaging *Frontiers in Physiology* Electrocardiographic Imaging **10** 145–62