This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

Block-Sparse Signal Recovery via General Total Variation Regularized Sparse Bayesian Learning

Aditya Sant*, *Student Member, IEEE,* Markus Leinonen†, *Member, IEEE,* and Bhaskar D. Rao*, *Fellow, IEEE*

Abstract*—One of the main challenges in block-sparse signal recovery, as encountered in, e.g., multi-antenna mmWave channel models, is block-patterned estimation** *without knowledge of block sizes and boundaries. We propose a novel Sparse Bayesian Learning (SBL) method for block-sparse signal recovery under unknown block patterns. Contrary to conventional approaches that impose block-promoting regularization on the signal components, we apply two classes of** *hyperparameter* **regularizers for the SBL cost function, inspired by total variation (TV) denoising. The first class relies on a conventional TV difference unit and allows performing the SBL inference iteratively through a set of convex optimization problems, enabling a flexible choice of numerical solvers. The second class incorporates a region-aware TV penalty to penalize the signal and zero blocks in a dissimilar manner, enhancing the performance. We derive an alternating optimization algorithm based on expectation-maximization to perform the SBL inference through computationally efficient parallel updates for both the regularizer classes. The numerical results show that the proposed TV-regularized SBL algorithm is robust to the nature of the block structure and is capable of recovering signals with both block-patterned and isolated components, proving effective for various signal recovery systems.**

***Index terms*—Compressed Sensing, Block-sparsity, Sparse Bayesian Learning, Total Variation, Expectation-maximization, Majorization-minimization, Cyclic Optimization.**

I. Introduction

Solving an underdetermined system of linear equations has been heavily studied in the signal processing literature. In many signal acquisition and estimation tasks, the underlying signals are *sparse.* Compressed Sensing (CS) theory [2], [3] has extensively studied stability conditions, algorithms, and convergence for sparse signal estimation. Within the CS paradigm, one particular signal class is *block-sparse* signals where consecutive groups of elements are alternatively nonzero or zero. Block-sparse signal recovery has various applications in wireless communication, audio, and image processing. Our primary interest resides in mmWave channel estimation and modeling where the received signal consists of angular multipath components that impinge on the receive antenna as clustered rays (hence block-sparse) [4]–[6].

*Aditya Sant and Bhaskar D. Rao are with Department of Electrical and Computer Engineering, University of California San Diego. e-mail: {ausant, brao}@eng.ucsd.edu.

†Markus Leinonen is with Centre for Wireless Communications – Radio Technologies, University of Oulu, Finland. e-mail: markus.leinonen@oulu.fi.

The work of A. Sant and B. D. Rao has been financially supported in part by ONR Grant No. N00014-18-1-2038, NSF grant CCF-2124929 and the UCSD Center for Wireless Communications. The work of M. Leinonen has been financially supported in part by Walter Ahlström Foundation through Tutkijat Maailmalle program, Infotech Oulu, the Academy of Finland (grant 323698 and 319485), and Academy of Finland 6Genesis Flagship (grant 318927).

A portion of this work, particularly the ideas in Section III-A, appear at the ICASSP 2021 [1].

The key challenge with block-sparse recovery is to model inter-element dependency, in addition to the sparsity constraint. We address the most general class of block-sparse signals: block boundaries are *unknown* and the block sizes are, in general, *unequal*. In this setup, the number of possible block combinations involved in the search grows exponentially with the signal length. Imposing all signal boundaries without prior structure and knowledge would prevent scaling to large signal sizes and thus prove inefficient for recovery. On the other hand, employing a conventional CS algorithm with no block structure regularization would clearly underperform by not taking the full advantage of the signal structure. Thus, there is an inherent *trade-off* in incorporating a *prior* signal structure in block-sparse recovery under unknown block boundaries: pre-defining the search grid enables accurate modeling at the cost of excessive complexity, whereas computationally tractable, flexible modeling of block-sparse priors may hamper the ability to recover arbitrary block structures. This calls for the design of a computationally efficient signal recovery algorithm that imposes an effective, yet *robust* prior to model the underlying non-uniform block-sparsity, which is the main focus of our paper.

A. Overview of Block-Sparse Signal Recovery

Block-sparse recovery methods have enforced block-patterned structures through block partitioning or efficient inter-element coupling. The early attempts assumed known block sizes and modified conventional CS algorithms to support block-sparse signal recovery. These include Group-Lasso [7], Group Basis Pursuit [8], Model-based CoSaMP [9], Block-OMP [10], and a block ℓ_2-norm based method [11]. Extending to the case of unknown block partitions, algorithms such as Struct-OMP in [12] and the method based on graphical models in [13] were developed. The approach in [13] uses the Boltzmann Machine model to capture inter-element dependencies; the method suffers from high complexity, inhibiting the scaling to large dictionaries and signal dimensions.

Apart from conventional CS approaches, Sparse Bayesian Learning (SBL) [14], [15] has shown superior performance for block-sparse recovery, especially in multiple measurement vector (MMV) scenarios. The first work is [16], where the developed block Sparse Bayesian Learning (BSBL) algorithm assumes known block partitions and models temporal signal correlations in an MMV-SBL problem. The work in [17] proposes different optimization methods for the BSBL inference, including the extension to unknown block structures, which however does not follow an elegant optimization framework.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

Differently, [18] uses Bayesian compressive sensing and incorporates a spike-and-slab prior to model both block-patterned and individual sparsity. The inference in [18] relies on time-consuming Gibbs Sampler and Markov Chain Monte Carlo (MCMC) methods.

The Pattern-coupled SBL (PC-SBL) method [19] initiated a fresh view to incorporate block-sparse structures by coupling the underlying SBL *hyperparameters*. The Non-uniform Burst Sparsity algorithm in [20] improved on PC-SBL inference by using the Variational Bayesian Inference [21]. Coupled priors for block-sparse signal recovery are also used in the Extended-BSBL (EBSBL) method in [22]. Unlike the PC-SBL algorithm, EBSBL algorithm gives an equal weight to the neighbouring parameters, leading to performance superior to BSBL, but inferior to PC-SBL.

All the above coupled-priors-based algorithms incorporate a hyperprior in the SBL parameter space (see, e.g., Gamma hyperprior in [14]), which needs to be specifically tuned for a particular block-sparse signal structure. Instead of such offline tuning, the hyperparameters could be estimated jointly with the signal. Such an approach is proposed in [23], where the signal and support random variables are modeled via a Gaussian-Bernoulli prior whose hyperparameters are also estimated from the observed data. Since exact update equations are no longer tractable for such models, the MCMC method is used for parameter estimation. Based on a clustered sparsity model, a hybrid clustered sparse prior is introduced in [24]. This framework extends PC-SBL [19] by considering the PC-SBL's coupling coefficient as a hyperparameter which is then adaptively adjusted based on the observed data.

B. Main Contribution: Total Variation Regularizers for SBL

We address compressed estimation of block-sparse signals in an MMV setup. We propose a novel SBL hyperparameter prior/regularizer to encourage block-patterned sparsity where the underlying sparsity profile is non-uniform and *unknown*, i.e., without any prior knowledge of block sizes and boundaries. To this end, we introduce two classes of *Total Variation* (TV) inspired regularizers that promote contiguous signal and zero regions by enforcing low TV in the hyperparameter space of SBL. The salient features of these regularizers, and the underlying optimization used, are given below:

- The first regularizer class, *Conventional TV Regularizers*, uses the standard TV of the form $T(\mathbf{x}) = \sum_i |x_i - x_{i-1}|$ (i.e., the absolute difference between two consecutive elements) and its variants utilizing sparsity promoting regularizers from the CS domain on the TV penalty. This enforces minimal hyperparameter variation in both the signal and zero regions of a block-sparse signal.
- The second regularizer class, *Region-aware Regularizers*, introduces more robust regularization that penalizes the signal and zero regions differently; this region-awareness stems from the incorporation of a nonlinear hyperparameter transformation $g(\cdot)$, thus creating a general version of the TV as $T(\mathbf{x}) = \sum_i |g(x_i) - g(x_{i-1})|$.
- A majorization-minimization approach is utilized to derive iterative convex solvers for the class of Conventional TV Regularizers. We have analyzed this framework separately in our prior work [1].
- We develop an iterative algorithm for the Expectation-Maximization (EM) based SBL inference using cyclic optimization, tailored for Region-aware Regularizers, and show its convergence properties. The procedure decouples the inference into parallel updates for each hyperparameter component, establishing a computationally efficient optimization method. We further unify the method to build a universal inference strategy for the TV-regularized SBL under both regularization classes.

Numerical experiments show that by inducing a soft, flexible TV prior, the proposed TV-regularized SBL method is *robust* to sparsity structure; the algorithm attains definitive recovery from strict block-sparsity to fully random sparsity in spite of the block-pattern-inducing prior. We additionally note that the block-sparse signal generation is inspired by the real-world applications like mmWave channel modeling. To this end, we assume that the block-sparse signals do not have excessively large amplitude fluctuations within each signal block; the developed TV-SBL accommodates such homogeneous signals for resilient performance.

To the best of our knowledge, this is the first work to apply a TV-type penalty in the hyperparameter space of SBL for the purpose of encouraging block-sparsity. The framework is quite general and allows for more exploration of a wide range of regularizers, especially those inspired by CS.

Organization: The paper is organized as follows. Section II formulates the block-sparse signal recovery problem through the SBL framework and introduces the two classes of TV regularization for block-sparsity. Section III presents a general optimization framework for the both regularizer classes. Section IV devises a universal EM-based alternating optimization method, establishes its proof of convergence, and outlines its algorithmic implementation. Numerical experiments are shown in Section V, and Section VI provides concluding remarks.

Notations: Vectors are denoted by lower-case boldface letters ($\mathbf{a}$) and defined as column vectors $\mathbf{a} = [a_1 \cdots a_N]^{\mathrm{T}}$. Matrices are denoted by upper-case boldface letters ($\mathbf{A}$). The vector at ith column of matrix $\mathbf{A}$ is denoted as $\mathbf{a}_i$. Transpose and Hermitian transpose are denoted as $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$, respectively. The trace of matrix $\mathbf{A}$ is denoted as $\mathrm{Tr}(\mathbf{A})$. A diagonal matrix with diagonal entries $a_1, \ldots, a_N$ is denoted by $\mathrm{diag}(a_1, \ldots, a_N)$. The default norm for a vector $||\mathbf{a}||$ is the 2-norm, unless otherwise specified. $||\mathbf{A}||$ denotes the Frobenius norm of matrix $\mathbf{A}$, unless otherwise specified. The ℓ_0-"norm" $\|\mathbf{a}\|_0$ counts the number of nonzero entries of vector $\mathbf{a}$. The notation $(\cdot)^{(k)}$ is used to denote the value at the kth iteration. We use $I(\cdot)$ as a shorthand for the standard indicator function for positive real numbers $\mathbf{1}_{\{\mathbb{R}^+\}}(\cdot)$. The statistical quantities $p(\mathbf{x}|\mathbf{y})$, $\mathbb{E}[\cdot]$, etc. have their usual meaning.

II. New Total Variation Regularizers for Block-Sparse Signal Recovery via SBL

A. Sparse Signal Recovery Problem

We consider a multiple measurement vector (MMV) problem where the objective is to simultaneously estimate L un-

known source vectors $\mathbf{x}_l \in \mathbb{C}^N$ with a common *block-sparse* structure from a collection of noisy linear measurements[1]

$$\mathbf{y}_l = \mathbf{A}\mathbf{x}_l + \mathbf{n}_l,\ l = 1, \ldots, L, \tag{1}$$

where $\mathbf{y}_l \in \mathbb{C}^M$ is a measurement vector at time instant l, $\mathbf{A} \in \mathbb{C}^{M \times N}$ is a fixed and known measurement matrix (or a basis matrix or dictionary), and $\mathbf{n}_l \sim \mathcal{CN}(\mathbf{0}, \lambda \mathbf{I})$ is a noise vector, independent of $\mathbf{x}_l$. Source vectors and noise vectors are assumed to be independent and identically distributed (i.i.d.) across the time instants.

Because of the block-sparse structure, $\mathbf{x}_l$ consists of successive signal blocks and zero blocks. The same sparsity pattern is shared among the collection of vectors $\{\mathbf{x}_l\}_{l=1}^L$, but the magnitudes of nonzero elements are drawn from an i.i.d. Gaussian distribution. Thus, the signal ensemble $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_L]$ is *block-row-sparse*. We consider the most general (and challenging) block-sparse inference task: both the block locations and (possibly non-uniform) sizes are *unknown*.

Prior to introducing our novel SBL-based method for block-sparse recovery, we provide a brief overview of the general SBL framework for sparse signal recovery.

B. SBL with Generalized Cost Function

SBL is inspired by the Automatic Relevance Determination mechanism from neural networks [25], [26], providing the means for relatively weighing the importance of different network weights, which could be sparse. There are various advantages for choosing SBL as a sparse signal recovery method for the MMV case:

- The M-SBL [27] parameter estimation abstracts each row of $\mathbf{X}$ by a single (hyper)parameter (γ_i), reducing the number of parameters to be estimated from NL to N compared to CS approaches.
- SBL falls into the category of *correlation-aware* methods which show superior sparse recovery performance [28].
- It has been shown in [15], [29] that the global minimum for the SBL cost function leads to maximally sparse solutions. Additionally, the number of local minima can be bounded above [15]; this bound is typically lower than that for conventional sparse recovery algorithms.
- SBL shows great promise for sparse signal recovery under correlated sources and ill-conditioned dictionaries [30], [31]. Such dictionaries (e.g., FFT bases) are often encountered in specific signal recovery applications, like wireless channel estimation.

We now describe the SBL inference. With an additive Gaussian noise model (1), the likelihood of the observations $p(\mathbf{y}_l|\mathbf{x}_l)$ is given by the Gaussian likelihood $\mathcal{CN}(\mathbf{A}\mathbf{x}_l, \lambda\mathbf{I})$. The SBL framework [27] assumes a prior distribution on $\mathbf{x}_l$; for each signal $\mathbf{x}_l \in \mathbb{C}^N$, $l = 1, \ldots, L$, this is characterized by a parametric Gaussian distribution $\mathcal{CN}(\mathbf{0}, \mathbf{\Gamma})$ as

$$p(\mathbf{x}_l; \boldsymbol{\gamma}) = \tfrac{1}{|\pi\mathbf{\Gamma}|} \exp\left(-\mathbf{x}_l^{\mathrm{H}} \mathbf{\Gamma}^{-1} \mathbf{x}_l\right), \tag{2}$$

where $\boldsymbol{\gamma} = [\gamma_1 \cdots \gamma_N]^{\mathrm{T}} \in \mathbb{R}_+^N$ is a vector of *hyperparameters*, adjusting the variance of each signal component $x_{l,i}$, $i = 1, \ldots, N$, and $\mathbf{\Gamma} \triangleq \mathrm{diag}(\gamma_1, \ldots, \gamma_N)$. The hyperparameter values $\boldsymbol{\gamma}$ reflect the sparsity profile of the block-row-sparse $\mathbf{X}$; a suitable prior on $\boldsymbol{\gamma}$ can lead $\mathbf{x}_l$ to model many interesting sparse priors, e.g., Gaussian scale mixtures [14], [32], [33].

The posterior density $p(\mathbf{x}_l|\mathbf{y}_l; \boldsymbol{\gamma})$ is also Gaussian as $\mathcal{CN}(\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}, \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}})$, where

$$\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}} = \lambda^{-1} \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}} \mathbf{A}^{\mathrm{H}} \mathbf{y}_l,\ \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}} = \left(\lambda^{-1} \mathbf{A}^{\mathrm{H}} \mathbf{A} + \mathbf{\Gamma}^{-1}\right)^{-1}. \tag{3}$$

For a given $\boldsymbol{\gamma}$, the estimate of each signal $\{\mathbf{x}_l\}_{l=1}^L$ is formed as $\hat{\mathbf{x}}_{l,\mathrm{SBL}} = \boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}$ according to (3). Following [27], the hyperparameter estimation is done through Type-II Maximum a Posterior (MAP) estimation over $\boldsymbol{\gamma}$ as

$$\begin{aligned} \boldsymbol{\gamma}^* &= \underset{\boldsymbol{\gamma} \succeq \mathbf{0}}{\mathrm{argmax}}\ \log\, p(\boldsymbol{\gamma}|\mathbf{y}_1, \ldots, \mathbf{y}_L) \\ &= \underset{\boldsymbol{\gamma} \succeq \mathbf{0}}{\mathrm{argmin}}\ L \log|\mathbf{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^L \mathbf{y}_l^{\mathrm{H}} \mathbf{\Sigma}_{\mathbf{y}}^{-1} \mathbf{y}_l - \log p(\boldsymbol{\gamma}), \end{aligned} \tag{4}$$

where $\mathbf{\Sigma}_{\mathbf{y}} = \lambda\mathbf{I} + \mathbf{A}\mathbf{\Gamma}\mathbf{A}^{\mathrm{H}}$ is the measurement model covariance matrix and $\log p(\boldsymbol{\gamma})$ is the *hyperprior* on $\boldsymbol{\gamma}$. The expression in (4) is the generalized *MMV-SBL cost function*.

Two important remarks regarding (4) are in order.

1) *Role of the prior:* A majority of works consider so-called *non-informative* (i.e., uniform) prior $\log p(\boldsymbol{\gamma})$, which corresponds to type-II Maximum Likelihood estimation above. However, in signal processing tasks, where one possesses some *prior information* of the structure of signal $\mathbf{X}$, an appropriate choice of an *informative* prior of hyperparameter $\boldsymbol{\gamma}$ can significantly improve the inference performance [34]–[36]. We address the latter by incorporating the knowledge of the underlying block-sparse signal structure into the hyperprior $\log p(\boldsymbol{\gamma})$.
2) *Minimization strategies:* The SBL cost function (4) is, in general, *non-convex* in $\boldsymbol{\gamma}$ due to the concave term $\log|\mathbf{\Sigma}_{\mathbf{y}}|$; convexity of $\log p(\boldsymbol{\gamma})$ depends on the prior. Different minimization strategies for (4) have been proposed, along with their convergence guarantees. To this end, we develop and discuss specific optimization strategies tailored for our novel SBL method in Sec. III.

C. SBL with Novel TV-based Regularizers

Regarding the MMV-SBL cost function (4), this section presents our novel SBL priors, i.e., *hyperparameter regularizers*, that will encourage block-sparse signal recovery. We introduce two regularization classes inspired by **Total Variation (TV)** and various CS-based sparse regularizers. TV has been used extensively in image processing for regularization and denoising [37]–[39]. TV regularization has also been used for group-sparse recovery along with the sparsity-inducing regularizers (like ℓ_1-norm penalty) in such image recovery tasks [40]–[45]. These approaches enforce structure

[1]Since our framework is not application-specific, we call each unknown $\mathbf{x}_l$ a source for convenience. The considered MMV setting arises in, e.g., wireless communications: 1) direction of arrival estimation of clustered multipath components at mmWave frequencies, where the block-sparse $\mathbf{x}_l$ represents a vector of beam space angular components of the channel for time instant l, and 2) a user activity detection and channel estimation problem in multi-antenna communications under uncorrelated fading channels with spatially correlated activity patterns, where the block-sparse $\mathbf{x}_l$ is the channel vector associated with N users at time instant l.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

by working directly on signal $\mathbf{x}_l$ – a more challenging and less efficient approach for the complicated block-sparsity problem.

Differently, our approach relies on enforcing special structures on the hyperparameter vector $\boldsymbol{\gamma}$. It is noteworthy that imposing the regularizer on the hyperparameters rather than the source vectors $\mathbf{x}_l$ is an important distinction and also key to the success of our approach. Block-sparse recovery algorithms have been developed introducing priors on the hyperparameter space (see Sec. I-A). The BSBL [17] algorithm is tailored to fixed block sizes, thus limiting its utility for more flexible block-sparse recovery. The algorithms using rigid SBL hyperparameter coupling [19], [20], although highly effective for flexible block-sparse recovery, are herein shown to be sensitive to recovering signals with both block-sparse and isolated sparse components; these stronger priors are biased to block structures in the underlying signal. We aim to bridge this gap by introducing a softer TV-inspired prior $p(\boldsymbol{\gamma})$ on the hyperparameter space. This circumvents excessive coupling bias on the signal and makes the proposed method extremely *robust* and capable of accommodating both block-sparse and isolated sparse components. Empirical evidence for our claims is provided through the simulation results for different classes of block-sparse signals in Sec. V.

Consider the MMV-SBL cost function (4). We denote the hyperprior as $\tilde{\beta}\, T(\boldsymbol{\gamma}) \triangleq -\log p(\boldsymbol{\gamma})$, where $\tilde{\beta}$ is a non-negative weighting parameter controlling the emphasis on the prior, and $T(\cdot)$ is a general TV-type penalty of vector $\boldsymbol{\gamma}$, which will become explicit in sequel. Accordingly, we will minimize the *TV-regularized MMV-SBL cost function*

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L} \mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \tilde{\beta}\, T(\boldsymbol{\gamma}). \quad (5)$$

Regardless of the form of $T(\cdot)$, we refer to our proposed method collectively as **TV-SBL**.

Recall from (4) that $\tilde{\beta}$ in (5) serves as the interface on how much weight the data $\mathbf{Y}$ carries in (the optimization over) the posterior distribution versus the weight carried by the prior information about the signal structure, captured by $T(\boldsymbol{\gamma})$. This brings us to motivate the choice of an appropriate $T(\cdot)$. The first thing to stress is that the purpose of the regularization $T(\cdot)$ is not to promote sparse solutions per se. Namely, in the canonical part of the SBL cost function in (5), the log det term $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ takes the role of imposing a sparsity penalty via the principles of automatic relevance determination. Thus, the role of $T(\cdot)$ is to *augment* the SBL's objective to encourage *special* sparse structures, herein, block-sparse structures.

Conventional CS techniques induce sparsity via minimization of the ℓ_0-"norm" $||\mathbf{x}||_0 = \sum_{i=1}^{N} I(|x_i|)$. Many surrogate measures have been used, the most common one being the ℓ_1-norm in CS [46], [47]. We harness this indicator function perspective to promote block-sparsity. By extending this element-counting function to a *block-counting function*, we introduce two novel classes of regularizers $T(\cdot)$: *(i) Conventional TV Regularization* and *(ii) Region-aware Regularization using TV*. We introduce three analytical TV-SBL regularizers – Linear TV, Log TV and DoL TV – with the first two belonging to the conventional TV regularization class, and the third to the region-aware regularization class. We elaborate further below.

Conventional TV Regularization

Our initial approach defines a block (either signal or zero block) as a region with constant γ_i's, analogous to the BSBL algorithm in [16], yet without the knowledge of the true block distribution. We define the related regularizer using the block-counting function that is equal to TV on $\boldsymbol{\gamma}$, i.e.,

$$T(\boldsymbol{\gamma}) = \textstyle\sum_{i=2}^{N} I(|\gamma_i - \gamma_{i-1}|). \quad (6)$$

Using equal variances for the entries within a block, the measure (6) counts the number of edges in the underlying block structure. Note that even if the regularizer (6) tends to enforce equal values of γ_i's for the consecutive nonzero entries, this does not stringently translate to restraining the corresponding signal entries $x_{l,i}$ to have equal magnitudes. This is the main asset in regularizing the hyperparameters, not the signal magnitudes. Thus, while simple, the imposed hierarchical prior structure admits significant boost in recovering various block-sparse signals, as demonstrated by the numerical results in Sec. V.

Armed with the ideal measure (6), we can use *tractable measures* developed in CS on the TV input variable $|\gamma_i - \gamma_{i-1}|$ to identify appropriate block structures. CS theory has developed many regularizers that are monotonically increasing and concave on the positive orthant to promote sparsity. We investigate two such surrogate functions, introducing our two conventional TV regularizers addressed in this paper.

1) Linear TV: The linear TV regularizer is equivalent to the ℓ_1 penalty in CS and is given by the form

$$T(\boldsymbol{\gamma}) = \textstyle\sum_{i=2}^{N} |\gamma_i - \gamma_{i-1}|. \quad (7)$$

As stated earlier, TV has also been used in different signal processing applications to preserve edges and enforce local smoothness. We use this convex regularizer to enforce a block structure in the recovered signal. In addition to the signal regions, this penalty is found to "denoise" the zero regions more effectively than the unregularized SBL algorithm [1].

2) Log TV: Another widely used regularizer in CS is $\sum_{i=1}^{N} \log(|x_i| + \epsilon)$, where ϵ is a positive stability parameter. This log-sum regularizer employs an iterative reweighted ℓ_1 minimization algorithm and has been shown to yield superior recovery [47], [48]. Utilizing this regularizer for block-sparsity, the log TV regularizer is given by

$$T(\boldsymbol{\gamma}) = \textstyle\sum_{i=2}^{N} \log(|\gamma_i - \gamma_{i-1}| + \epsilon). \quad (8)$$

As in the CS literature, the log TV based approach is found to be more effective than the linear TV, using the MM optimization framework (see Sec. III-A) [1]. This is due to its better resemblance to ℓ_0-"norm" [47], allowing more signal variance differences within a block and restraining small (faulty) signal estimate components to emerge.

The block-counting function (6) enforces blocks quite rigidly through constant γ_i values, irrespective of a block being a zero or nonzero block. A more flexible block-counting function, explained next, incorporates this crucial difference between zero and nonzero blocks.

Region-aware Regularization using TV (RAR TV)

We introduce a new terminology here, namely, zero and

TABLE I
COMPARISON OF THE KEY PROPERTIES FOR THE TV-SBL REGULARIZERS (7), (8), AND (11)

Property	Linear TV (7)	Log TV (8)	DoL TV (11)
Block-counting function	Conventional TV (6)	Conventional TV (6)	Region-aware using TV (9)
Analogous CS regularizer	ℓ_1-norm regularization	log-sum regularization	No conventional CS regularizer
Block structure enforcement	Minimize γ_i variation	Minimize γ_i variation	Minimize γ_i zero–nonzero transition
Block type symmetry	Invariant to block type	Invariant to block type	Differential zero–nonzero γ_i regularizer
Convex optimization solution	Possible through MM	Possible through MM	Not possible (EM-based optimization used)
Performance	Weakest TV-SBL regularizer	Stronger than Linear TV	Strongest TV-SBL regularizer

signal *regions*. Inspired by a pictorial representation of block-sparse signals (e.g., block-sparse angular regions for mmWave channels), a signal region – as a broader description of a signal block – consists of a contiguous sequence of nonzero entries, which are not necessarily associated with the same γ_i value. We have empirically observed that there exists a dissimilarity in the recovery of zero and signal regions of the sparse vector $\mathbf{x}_l$. We modify our original block-counting function (6) to incorporate this differentiation as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |I(\gamma_i) - I(\gamma_{i-1})|. \tag{9}$$

The block-counting function (9), unlike (6), will minimize the number of transitions between a signal region and a zero region, while allowing arbitrary variation of signal magnitudes within the blocks of the signal region. Further, this will enforce strict contiguous zero blocks; hence the name *Region-aware Regularization (RAR)*. This differential treatment of the zero and nonzero regions translates to a nonlinear scaling of the hyperparameter components. More explicitly, we use a tractable relaxation of (9) as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |g(\gamma_i) - g(\gamma_{i-1})|, \tag{10}$$

where $g(\cdot)$ is a non-decreasing concave function in the positive quadrant, which (relatively) amplifies lower γ_i values while scaling down the higher values corresponding to large signal components. We term the penalty in (10) as the *Difference of Functions (DoF)* regularizer. We use this form, with its concave non-decreasing behavior, for the optimization analysis of TV-SBL in Sec. IV. Prior to this, we introduce one specific realization of $g(\cdot)$ for the DoF penalty, which will be used to illustrate the utility of the RAR strategy.

3) Difference of Logs (DoL) TV: Following the above arguments for the utility of the $\log(\cdot)$ transform in CS as surrogate measures of the ℓ_0-"norm", we propose a novel *Difference of Logs (DoL)* TV penalty as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |\log(\gamma_i) - \log(\gamma_{i-1})|. \tag{11}$$

Here, the role of the $\log(\cdot)$ function[2] is to nonlinearly scale the hyperparameter vector $\boldsymbol{\gamma}$ to differentially treat the zero and signal regions, which is the main feature required for $g(\cdot)$, as motivated above. We remark that other tractable surrogate functions $g(\cdot)$ that better resemble the indicator function $I(\cdot)$ in (9) are worth studying as future work.

[2]Some of our experiments (not provided herein) have shown that the concave square root penalty $g(\gamma_i) = \sqrt{\gamma_i}$ is another potential variant for (10), performing similar to the $\log(\cdot)$ penalty.

Remark 1. One feature of interest here is that the penalty $|\log(\gamma_i) - \log(\gamma_{i-1})|$ in (11) alternatively reduces the TV in the hyperparameter space using *the ratio* $\frac{\gamma_i}{\gamma_{i-1}} \to 1$. The analysis of the TV reduction by matching the ratio to 1 warrants further elaboration and discussion, and it is left for future work.

Remark 2. The value of regularization weight parameter β has clearly a significant impact on the TV-SBL's performance and needs thus appropriate adjustment. As for any regularization-based method, having a systematic way of defining the optimal β is elusive. Pragmatically, if the estimate of $\boldsymbol{\gamma}$ and, consequently, the estimate of $\mathbf{x}_l$ tend to exclusively show isolated sparse components for a block-sparse $\mathbf{X}$, increasing β promotes the blocks to emerge. On the contrary, if the estimate vectors are non-sparse with overly smooth envelopes, β should be decreased. We remark that configuring β could also be incorporated into a learning framework. Specific tuning of β is outside of the scope of this work.

Through this section, we have introduced three novel block-sparsity-inducing regularizers: (7), (8), and (11). The key properties highlighting the differences between these regularizers are summarized in Table I. We now move on to analyze the optimization of the TV-regularized MMV-SBL cost function (5) under each regularizer.

III. OPTIMIZATION FRAMEWORK FOR TV-SBL

SBL optimization has been extensively studied in various contexts. The original approach was proposed by Tipping [14], using fixed point iteration. The most popular algorithm for SBL optimization is the Expectation-Maximization (EM) algorithm, as used in [15], [16], [27]. Another common approach is the Majorization-Minimization (MM) framework [1], [49] that relaxes the non-convex optimization problem (4) into a sequence of convex optimization problems. Other strategies include the iterative reweighted ℓ_1 and ℓ_2 methods [32] and Generalized Approximate Message Passing (GAMP) [31].

The fundamental difference in the underlying block-counting measures between the Conventional TV in (6) and RAR TV in (9) classifies the optimization procedures for the TV-SBL cost (5) under their respective labels.

1) **Conventional TV Optimization:** The linear and log TV penalties in (7) and (8) can be tackled by convex optimization tools after appropriately *relaxing* the problem (5), e.g., using the MM framework. We address such MM-based optimization in Sec. III-A.
2) **RAR TV Optimization:** The DoL penalty in (11) precludes the use of convex solvers, as will be discussed

in Sec. III-B. We approach this by splitting the overall optimization into two sequential sub-problems, and combining the solutions. We use the EM-based inference to generate closed-form update expressions. The EM preliminaries are presented in Sec. III-C and our overall optimization procedure is detailed in Sec. IV.

Both these frameworks reformulate the original TV-SBL cost (5), to be handled by different optimization tools.

A. Conventional TV-SBL through Convex Optimization

We apply the MM approach and derive an iterative algorithm for minimizing the TV-SBL cost in (5) under the linear TV and log TV regularizer in (7) and (8), respectively.

1) Linear TV: The TV-SBL optimization (5) for the linear TV regularizer in (7) is

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\, L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \tilde{\beta}\,\sum_{i=2}^{N}|\gamma_i-\gamma_{i-1}|. \tag{12}$$

Following [49], we majorize (i.e., linearize) the concave term $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ by its first-order Taylor approximation at point $\boldsymbol{\Gamma}^{(j)}$ and solve at iteration j a convex optimization problem

$$\begin{aligned}\boldsymbol{\gamma}^{(j+1)} = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\,\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}\right)^{-1}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}\right)\\ &+\textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \tilde{\beta}\sum_{i=2}^{N}|\gamma_i-\gamma_{i-1}|,\end{aligned} \tag{13}$$

and then update the measurement covariance matrix $\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}$ using the newly obtained $\boldsymbol{\gamma}^{(j+1)}$. Any convex solver can be employed to solve (13).

2) Log TV: The TV-SBL optimization (5) for the log TV regularizer in (8) is

$$\begin{aligned}\boldsymbol{\gamma}^* = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l\\ &+\tilde{\beta}\textstyle\sum_{i=2}^{N}\log(|\gamma_i-\gamma_{i-1}|+\epsilon).\end{aligned} \tag{14}$$

The MM approach for this cost function is similar to (12), but we additionally majorize the concave log TV penalty by its first-order Taylor approximation at points $(\gamma_i^{(j)}-\gamma_{i-1}^{(j)})$, $i=2,\ldots,N$. Thus, at iteration j, we solve the convex problem

$$\begin{aligned}\boldsymbol{\gamma}^{(j+1)} = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\,\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}\right)^{-1}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}\right)\\ &+\textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \tilde{\beta}\sum_{i=2}^{N}\dfrac{|\gamma_i-\gamma_{i-1}|}{|\gamma_i^{(j)}-\gamma_{i-1}^{(j)}|+\epsilon},\end{aligned} \tag{15}$$

followed by updating $\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}$ using the newly obtained $\boldsymbol{\gamma}^{(j+1)}$. Hence, even though the log TV penalty is not convex, the majorization (15) finds an upper-bound solution to (14) via convex solvers. Implementation via common solvers such as CVX [50] is detailed in our earlier work [1].

B. Difficulty in Majorizing DoL TV Penalty

Consider the DoL penalty in (11). The term $|\log(\gamma_i)-\log(\gamma_{i-1})|$ is concave in γ_i for $\gamma_i>\gamma_{i-1}$ and convex in γ_i for $\gamma_i<\gamma_{i-1}$. Thus, unlike the linear and log TV in (12) and (14) respectively, we cannot find a universal majorizer for the DoL function. In fact, this holds true for any DoF TV penalty in (10) with a concave function $g(\cdot)$. Until further notice, we proceed with the DoF regularizer (10) while keeping in mind that its special case $g(\cdot)=\log(\cdot)$ realizes the DoL in (11).

The TV-SBL optimization (5) under the DoF TV penalty in (10) reads as

$$\begin{aligned}\boldsymbol{\gamma}^* = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l\\ &+\tilde{\beta}\textstyle\sum_{i=2}^{N}|g(\gamma_i)-g(\gamma_{i-1})|.\end{aligned} \tag{16}$$

We solve (16) with the EM method. In particular, as it will be seen in Sec. IV, this converts the minimization (16) into a form that can be solved efficiently using cyclic optimization. Before elaborating further, we describe the general EM procedure used in the TV-SBL optimization.

C. TV-SBL Inference through Expectation-Maximization

As opposed to a single-shot minimization (16), the EM algorithm finds the estimate of the hyperparameter vector $\boldsymbol{\gamma}$ iteratively. We follow the EM-SBL framework, introduced in [15] and extended to MMV in [27]. Using the standard EM theory [51], $\mathbf{Y}$ is the observation variable, $\mathbf{X}$ is the hidden variable and $\boldsymbol{\gamma}$ is the unknown parameter to be estimated. Each of the variables $\mathbf{Y},\mathbf{X}$, and $\boldsymbol{\gamma}$ is as determined in (1) and (2).

E-step

Treating $(\mathbf{Y},\mathbf{X})$ as the complete data and using the Markov chain $\boldsymbol{\gamma}\to\mathbf{X}\to\mathbf{Y}$, the joint distribution gives

$$\log p(\mathbf{X},\mathbf{Y},\boldsymbol{\gamma}) = \log p(\mathbf{Y}|\mathbf{X}) + \log p(\mathbf{X}|\boldsymbol{\gamma}) + \log p(\boldsymbol{\gamma}). \tag{17}$$

The Q-function is evaluated by averaging out the hidden variable $\mathbf{X}$ as

$$\begin{aligned}Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y},\mathbf{X},\boldsymbol{\gamma})\right]\\ &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y}|\mathbf{X})+\log p(\mathbf{X}|\boldsymbol{\gamma})+\log p(\boldsymbol{\gamma})\right]\\ &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y}|\mathbf{X})+\log\textstyle\prod_{l=1}^{L}p(\mathbf{x}_l|\boldsymbol{\gamma})\right]\\ &\quad+\log p(\boldsymbol{\gamma}),\end{aligned} \tag{18}$$

where the expectation is with respect to the posterior $p(\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}) = \prod_{l=1}^{L}p(\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)})$. Removing the terms that do not depend on $\boldsymbol{\gamma}$, and using the result from [27] to evaluate the posterior expectation, we write the E-step as

$$\begin{aligned}Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) &= \textstyle\sum_{i=1}^{N}\Big[-L\log\gamma_i - \sum_{l=1}^{L}\dfrac{E_{l,i}^{(k)}}{\gamma_i}\Big] + \log p(\boldsymbol{\gamma})\\ &= L\Big\{\textstyle\sum_{i=1}^{N}\Big[-\dfrac{E_i^{(k)}}{\gamma_i}-\log\gamma_i\Big] - \dfrac{\tilde{\beta}}{L}T(\boldsymbol{\gamma})\Big\},\end{aligned} \tag{19}$$

where, using (3), we defined the quantities for iteration k as

$$\begin{aligned}E_i^{(k)} &= \frac{1}{L}\textstyle\sum_{l=1}^{L}E_{l,i}^{(k)}\\ &= \frac{1}{L}\textstyle\sum_{l=1}^{L}\Big[\left[\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)}}(i)\right]^2 + \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}^{(k)}}(i,i)\Big].\end{aligned} \tag{20}$$

Remark 3. For the subsequent analysis, we use the regularization normalized by the number of snapshots, i.e., $\beta \triangleq \tilde{\beta}/L$.

Specializing (19) to the DoF TV regularizer in (10), we re-define the E-step cost function as

$$J^{(k)}(\boldsymbol{\gamma}) \triangleq \textstyle\sum_{i=1}^{N} f_i^{(k)}(\gamma_i) + \beta\,\sum_{i=2}^{N}|g(\gamma_i)-g(\gamma_{i-1})|, \tag{21}$$

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

$$\begin{aligned} J^{(k)}(\boldsymbol{\gamma}) &= \textstyle\sum_{i\in\text{even}} \Big[f_i^{(k)}(\gamma_i) + \beta \ \big(\ \big|g(\gamma_i) - g(\gamma_{i-1})\big| + \big|g(\gamma_i) - g(\gamma_{i+1})\big| \ \big) \Big] + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i) \\ &= \textstyle\sum_{i\in\text{odd}} \Big[f_i^{(k)}(\gamma_i) + \beta \ \big(\ \big|g(\gamma_i) - g(\gamma_{i-1})\big| + \big|g(\gamma_i) - g(\gamma_{i+1})\big| \ \big) \Big] + \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i). \end{aligned} \tag{24}$$

where, for brevity, we introduced a function

$$f_i^{(k)}(\gamma_i) \triangleq \frac{E_i^{(k)}}{\gamma_i} + \log \gamma_i. \tag{22}$$

M-step

$$\begin{aligned} \boldsymbol{\gamma}^{(k+1)} &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmax}}\, Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) \\ &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\, J^{(k)}(\boldsymbol{\gamma}) \\ &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\, \textstyle\sum_{i=1}^{N} f_i^{(k)}(\gamma_i) + \beta \sum_{i=2}^{N} |g(\gamma_i) - g(\gamma_{i-1})|. \end{aligned} \tag{23}$$

Using EM, we have converted the original TV-SBL optimization from (16) to iterative updates as per (23). It is important to note that besides $J^{(k)}(\boldsymbol{\gamma})$ being non-convex in $\boldsymbol{\gamma}$, the TV term introduces *coupling* of the hyperparameters, preventing updates of $\{\gamma_i\}_{i=1}^N$ in parallel. Consequently, the computational complexity of vector optimization in (23) may become excessive for high-dimensional signal setups. Therefore, in the next section, we propose an alternating optimization framework that enables solving the M-step (23) efficiently via element-wise parallel updates.

IV. Alternating Optimization for Solving the M-Step of the EM-based TV-SBL

Our alternating optimization procedure for TV-SBL, detailed in this section, has been inspired by the coordinate descent optimization methods [52]–[56]. We extend these methods to handle our EM-based TV-SBL framework described in Sec. III-C. First, the detailed derivation is carried out for the M-step of DoF TV-SBL in (23) (for which the MM approach is not applicable). The section ends with establishing *a universal EM-based TV solver*, providing alternative solvers to the linear TV optimization (12) and log TV optimization (14) apart from their MM solutions in Sec. III-A.

A. Alternating Optimization and Convergence

In order to use the techniques from coordinate descent to solve (23), we first reformulate the cost function $J^{(k)}(\boldsymbol{\gamma})$ in (21). We begin by expressing the TV part $T(\boldsymbol{\gamma})$ as summations over the **even and odd indices**, as shown in (24). The form (24) rewrites the M-step optimization (23) as a summation over two disjoint sets of the elements of $\boldsymbol{\gamma}$, i.e., the even and odd index elements of $\boldsymbol{\gamma}$. This provides an ideal setting to use alternating optimization and separately optimize over the even and odd indices. For this alternating optimization procedure, we define the following function in relation to (24):

$$\begin{aligned} &F_i^{(k)}(\gamma_i; \gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}) \triangleq f_i^{(k)}(\gamma_i) + \\ &\beta \ \big(\big|g(\gamma_i) - g(\gamma_{i-1}^{(k)})\big| + \big|g(\gamma_i) - g(\gamma_{i+1}^{(k)})\big| \big), \end{aligned} \tag{25}$$

where $\gamma_{i-1}^{(k)}$ and $\gamma_{i+1}^{(k)}$ are fixed quantities at EM iteration k.

The following theorem presents our proposed EM-based alternating optimization method for DoF TV-SBL along with its convergence properties. In particular, the following update equations form the M-step (see (23)) tailored to TV-SBL.

Theorem 1. *For $F_i^{(k)}(\gamma_i; \gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)})$ defined by* (25)*, if the alternating optimization at iteration k is carried out as*

$$\begin{aligned} \gamma_i^{(k+1)} &= \underset{\gamma_i>0}{\operatorname{argmin}}\, F_i^{(k)}(\gamma_i; \gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}), \ \forall i \textit{ even} \\ \gamma_i^{(k+1)} &= \underset{\gamma_i>0}{\operatorname{argmin}}\, F_i^{(k)}(\gamma_i; \gamma_{i-1}^{(k+1)}, \gamma_{i+1}^{(k+1)}), \ \forall i \textit{ odd}, \end{aligned} \tag{26}$$

the cost function (21) *decreases over each iteration k, i.e.,*

$$J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq J^{(k)}(\boldsymbol{\gamma}^{(k)}). \tag{27}$$

Proof. The proof is given in Appendix A. □

By means of Theorem 1, we decouple the general DoF TV penalty and provide N separate update equations corresponding to each hyperparameter element γ_i. Moreover, by using this two-step alternating optimization over the even and odd indices of $\boldsymbol{\gamma}$, the M-step cost function (23) decreases over each iteration as per (27), which subsequently results in convergence to a local minimum of problem (23).

Owing to the generality of the cost function $J^{(k)}(\boldsymbol{\gamma})$ in terms of the choice of concave function $g(\cdot)$, it may not always be possible to find closed-form solutions to the minimization steps in (26). However, the following corollary asserts that for a convergent algorithm determined through (27), it is sufficient to obtain only a decrease in the optimization steps.

Corollary 1. *If the updated parameters $\boldsymbol{\gamma}^{(k+1)}$ satisfy*

$$\begin{aligned} &F_i^{(k)}(\gamma_i^{(k+1)}; \gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}) \leq F_i^{(k)}(\gamma_i^{(k)}; \gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}), \ \forall i \textit{ even}, \\ &F_i^{(k)}(\gamma_i^{(k+1)}; \gamma_{i-1}^{(k+1)}, \gamma_{i+1}^{(k+1)}) \leq F_i^{(k)}(\gamma_i^{(k)}; \gamma_{i-1}^{(k+1)}, \gamma_{i+1}^{(k+1)}), \\ &\qquad\qquad\qquad\qquad\qquad\qquad \forall i \textit{ odd}, \end{aligned}$$

then this is sufficient to imply (27)*.*

Proof. The proof follows from the steps similar to those for Theorem 1. In particular, the inequalities (36) and (38) do not require the obtained $\gamma_i^{(k+1)}$, $\forall i$, in (26) to be strict minimizers. □

Corollary 1 implies that we can use different optimization methods for (26), ranging from gradient descent to MM, to approximately solve the M-step of the TV-SBL. Having described the alternating optimization recipe for the TV-SBL, we now present its specific iterative algorithmic implementation.

B. Algorithm Implementation: Segment-wise Parallel Updates

This section elaborates the steps to solve each optimization sub-problem in (26) over the even and odd indices. We exploit the nature of the absolute value function $|\cdot|$ of the DoF regularizer to break each minimization here into individual segments, optimize these separately, and then combine the

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

results. Hence, we refer to our method as the *Segment-wise Parallel Update* algorithm. We begin with the update equation for the even indices in (26); the optimization for the odd indices will follow analogously.

We start by introducing quantities that order the neighbouring elements for each $\gamma_i^{(k)}$:

$$\gamma_{i,\max}^{(k)} \triangleq \max\big(\gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}\big),\ \gamma_{i,\min}^{(k)} \triangleq \min\big(\gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}\big). \quad (28)$$

Using (28), the optimization problem for the even indices in (26) reads as

$$\begin{aligned}\gamma_i^{(k+1)} &= \operatorname*{argmin}_{\gamma_i>0} F_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}) \\ &= \operatorname*{argmin}_{\gamma_i>0} \Big[f_i^{(k)}(\gamma_i) + \beta\big(\big|g(\gamma_i) - g(\gamma_{i,\max}^{(k)})\big| \\ &\qquad + \big|g(\gamma_i) - g(\gamma_{i,\min}^{(k)})\big| \big)\Big], \quad \forall i \text{ even}. \end{aligned} \quad (29)$$

The objective function of (29) is differentiable, except at points $\gamma_i \in \{\gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}\}$. To overcome the non-differentiability, we first define three mutually exclusive and collectively exhaustive *segments* on positive reals ($\mathbb{R}_+$):

$$\begin{aligned}\hat{\Gamma}_i &= \big\{\gamma_i \mid \gamma_i > \gamma_{i,\max}^{(k)} \geq 0\big\}, \\ \tilde{\Gamma}_i &= \big\{\gamma_i \mid \gamma_{i,\max}^{(k)} \geq \gamma_i \geq \gamma_{i,\min}^{(k)} \geq 0\big\}, \\ \bar{\Gamma}_i &= \big\{\gamma_i \mid \gamma_{i,\min}^{(k)} > \gamma_i \geq 0\big\}, \end{aligned} \quad (30)$$

wherein this function is continuous and differentiable. Using (30), the optimization problem (29) is equivalent to

$$\gamma_i^{(k+1)} = \operatorname*{argmin}_{\gamma_i \in \left\{\hat{\gamma}_i^{(k)}, \tilde{\gamma}_i^{(k)}, \bar{\gamma}_i^{(k)}\right\}} F_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}),\ \forall i \text{ even}, \quad (31)$$

where the three "*candidate*" solutions $\{\hat{\gamma}_i^{(k)}, \tilde{\gamma}_i^{(k)}, \bar{\gamma}_i^{(k)}\}$ are found by separate *segment-wise optimization* problems

$$\begin{aligned}\hat{\gamma}_i^{(k)} &\triangleq \operatorname*{argmin}_{\gamma_i \in \hat{\Gamma}_i} f_i^{(k)}(\gamma_i) + 2\beta g(\gamma_i) - \beta\big[g(\gamma_{i,\max}^{(k)}) + g(\gamma_{i,\min}^{(k)})\big], \\ \tilde{\gamma}_i^{(k)} &\triangleq \operatorname*{argmin}_{\gamma_i \in \tilde{\Gamma}_i} f_i^{(k)}(\gamma_i) + \beta\big[g(\gamma_{i,\max}^{(k)}) - g(\gamma_{i,\min}^{(k)})\big], \\ \bar{\gamma}_i^{(k)} &\triangleq \operatorname*{argmin}_{\gamma_i \in \bar{\Gamma}_i} f_i^{(k)}(\gamma_i) - 2\beta g(\gamma_i) + \beta\big[g(\gamma_{i,\max}^{(k)}) + g(\gamma_{i,\min}^{(k)})\big]. \end{aligned} \quad (32)$$

The constrained problems in (32) present our general approach to solve the TV-SBL problem for any general DoF TV penalty (10). However, these might be difficult to solve, especially through closed-form expressions. Fortunately, for certain controlled cases, like the DoL TV penalty (11), we can solve them equivalently, yet more efficiently, through the following two-step procedure: an *unconstrained update* followed by *projection* to the relevant segment.

Step 1: Unconstrained update

$$\begin{aligned}\hat{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) + 2\beta g(\gamma_i) \\ \tilde{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) \\ \bar{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) - 2\beta g(\gamma_i). \end{aligned} \quad (33)$$

Step 2: Segment-wise projection

$$\begin{aligned}\hat{\gamma}_i^{(k)} &= \max\big(\gamma_{i,\max}^{(k)}, \hat{\alpha}_i^{(k)}\big) \\ \tilde{\gamma}_i^{(k)} &= \min\big(\gamma_{i,\max}^{(k)}, \max\big\{\gamma_{i,\min}^{(k)}, \tilde{\alpha}_i^{(k)}\big\}\big) \\ \bar{\gamma}_i^{(k)} &= \max\big(0, \min\big\{\gamma_{i,\min}^{(k)}, \bar{\alpha}_i^{(k)}\big\}\big). \end{aligned} \quad (34)$$

Algorithm 1 Universal EM-Based Segment-wise Parallel Update Algorithm for TV-SBL

Input: $\mathbf{A}, \mathbf{Y}, \boldsymbol{\gamma}^{(0)}, \lambda, \beta$
Output: $\boldsymbol{\gamma}^{(k_{\max})}, \boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k_{\max})}}^{(k_{\max})}, \forall l$
for $k = 0$ to $k_{\max}$ **do**
  Evaluate $\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}^{(k)}$ and $\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)}}^{(k)}, \forall l$ using (3)
  Evaluate $E_i^{(k)}$ using (20)
  Evaluate $\hat{\alpha}_i^{(k)}$, $\tilde{\alpha}_i^{(k)}$, and $\bar{\alpha}_i^{(k)}, \forall i$ using Table II
  for $i \in$ even indices **do**
    Evaluate $\gamma_{i,\max}^{(k)}$ and $\gamma_{i,\min}^{(k)}$ using (28)
    Evaluate $\hat{\gamma}_i^{(k)}$, $\tilde{\gamma}_i^{(k)}$, $\bar{\gamma}_i^{(k)}$ using (34)
    Obtain $\gamma_i^{(k+1)}$ using (31)
  end for
  for $i \in$ odd indices **do**
    Evaluate $\gamma_{i,\max}^{(k+1)}$ and $\gamma_{i,\min}^{(k+1)}$ using (28)
    Evaluate $\hat{\gamma}_i^{(k)}$, $\tilde{\gamma}_i^{(k)}$, $\bar{\gamma}_i^{(k)}$ using (34)
    Obtain $\gamma_i^{(k+1)}$ using (31)
  end for
end for

Elaborate justification of this segment-wise parallel update optimization strategy is presented in Sec. IV-C.

As we mentioned earlier, the steps above are used to update the hyperparameter values in parallel, obtaining $\gamma_i^{(k+1)}$ for the even indices. Then, these updated values are used in the optimization over the odd indices through the optimization steps analogous to (28), (34), and (31).

C. Unifying TV-SBL using Alternating Optimization

We now specialize this framework and set up a *universal* TV-SBL solver to handle all the introduced TV regularizers: the linear TV (7), the log TV (8), and the DoL TV (11).

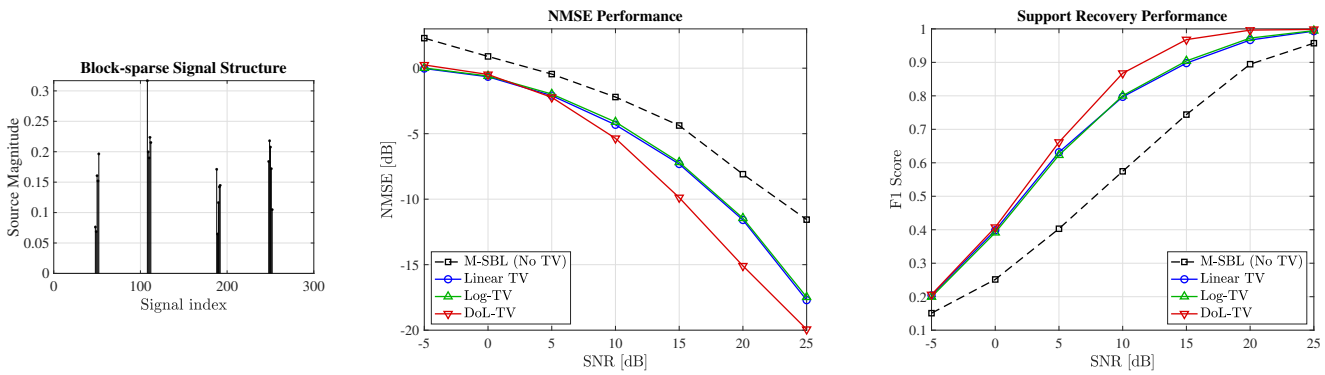
To begin, it is clear that for a convex cost (29), the segment-wise algorithm – following the steps (33), (34), and (31) – converges to a local minimum of (29), enjoying the convergence properties of Theorem 1, i.e., the TV-SBL M-step cost (23) decreases over each iteration. Since the DoL TV-SBL cost function (29) is non-convex, it is not immediately clear whether the unconstrained updates (33) in conjunction with the segment-wise projections in (34) work. To this end, Appendix B shows the segment-wise parallel updates for the DoL TV cost to be equivalent to (29), i.e., this optimization strategy finds a local minimum of the M-step cost (23). Thus, the DoL TV-SBL optimization, via the controlled nature of the DoL TV, adheres to the convergence results of Theorem 1, *despite the non-convexity*. Appendix B also derives the closed-form solutions to the unconstrained updates (33) for the DoL TV.

We derive an EM-based solution to the TV-SBL with the linear TV penalty (7) and the log TV penalty (8) as follows. For each penalty, we majorize the involved non-convex TV-SBL cost $F_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)})$ in (29) by a convex surrogate. Applying the segment-wise algorithm on the majorized convex cost function ensures the convergence of (23) according to Corollary 1, i.e., the TV-SBL M-step cost (23) decreases over each iteration. The closed-form solutions to the unconstrained

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

TABLE II
TV-SBL: UNCONSTRAINED UPDATES (33) FOR THE LINEAR, LOG, AND DOL TV PENALTIES

TV Penalty	$\hat{\alpha}_i^{(k)}$	$\tilde{\alpha}_i^{(k)}$	$\bar{\alpha}_i^{(k)}$
$\lvert\gamma_i - \gamma_{i-1}\rvert$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+2\beta}}$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}}}$	$\sqrt{\frac{E_i^{(k)}}{\max(\epsilon_0, q_i^{(k)}-2\beta)}}$
$\log(\lvert\gamma_i - \gamma_{i-1}\rvert + \epsilon)$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+\beta(a_i^{(k)}+b_i^{(k)})}}$	$\sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)}+\beta(b_i^{(k)}-a_i^{(k)})\right)}}$	$\sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)}-\beta(a_i^{(k)}+b_i^{(k)})\right)}}$
$\lvert\log(\gamma_i) - \log(\gamma_{i-1})\rvert$	$\frac{E_i^{(k)}}{1+2\beta}$	$E_i^{(k)}$	$\frac{E_i^{(k)}}{1-2\beta}$



(a) (b) (c)

Fig. 1. Performance of TV-SBL under the different TV penalties for $N = 300$, $M = 30$, and $L = 5$: (a) Homogeneous block-sparsity (4 blocks of length 5), (b) NMSE, and (c) Support recovery.

updates (33) for the linear and log TV under this majorization approach are derived in Appendix C and D, respectively.

Finally, our **universal EM-based TV-SBL algorithm** exploiting the segment-wise parallel updating is summarized in Algorithm 1; the required unconstrained update rules for (33) for each TV-penalty type are summarized in Table II.

D. Algorithm Complexity

Since the E-Step of TV-SBL is the same as the original M-SBL, the complexity of TV-SBL is similar to M-SBL. The main computational burden at each iteration of our proposed EM-based TV-SBL, as summarized in Algorithm 1, is the computation of the matrix inverse in the E-step in evaluating $\mathbf{\Sigma}_{\mathbf{x}_l|\mathbf{y}_l}^{(k)}$, $\forall l$, via (3). The number of floating-point operations is of order $\mathcal{O}(M^2N)$ per iteration k. It has been shown in [27] that the overall covariance computation can be done independent of the number of snapshots L. Thus, the overall order of the entire algorithm is $\mathcal{O}(k_{\max}M^2N)$, where $k_{\max}$ is the (maximum) number of iterations taken. For the M-step update (23) using the TV-SBL regularizers (7), (8), and (11), we have an $\mathcal{O}(1)$ parallel update procedure for each of the hyperparameter components $\boldsymbol{\gamma}_i$ (see Algorithm 1); thus faster computation than the E-step update (20), leaving the overall TV-SBL complexity as $\mathcal{O}(k_{\max}M^2N)$.

V. NUMERICAL RESULTS

We provide numerical results for the block-sparse signal recovery using the proposed TV-SBL algorithm, implemented via Algorithm 1. For the MMV setup (1), we consider a signal of length $N = 300$ with $M = 30$ measurements and $L = 5$ snapshots. We form the dictionary $\mathbf{A} \in \mathbb{R}^{M\times N}$ by first drawing its elements from a Gaussian distribution, and then normalizing the columns as $\|\cdot\|_2 = 1$. The signal ensemble $\mathbf{X}$ contains K nonzero rows and each nonzero element is drawn from $\mathcal{N}(0, 1/K)$. We consider three classes of sparse signal distributions:

1) *Homogeneous* block-sparse signal with total sparsity $K = 20$, composed of 4 blocks of length 5 each;
2) *Random* sparse signal with $K = 15$ randomly placed nonzero components, which are thus mostly isolated;
3) *Hybrid* sparse signal with total sparsity $K = 20$, composing 3 blocks of length 5, and 5 isolated components.

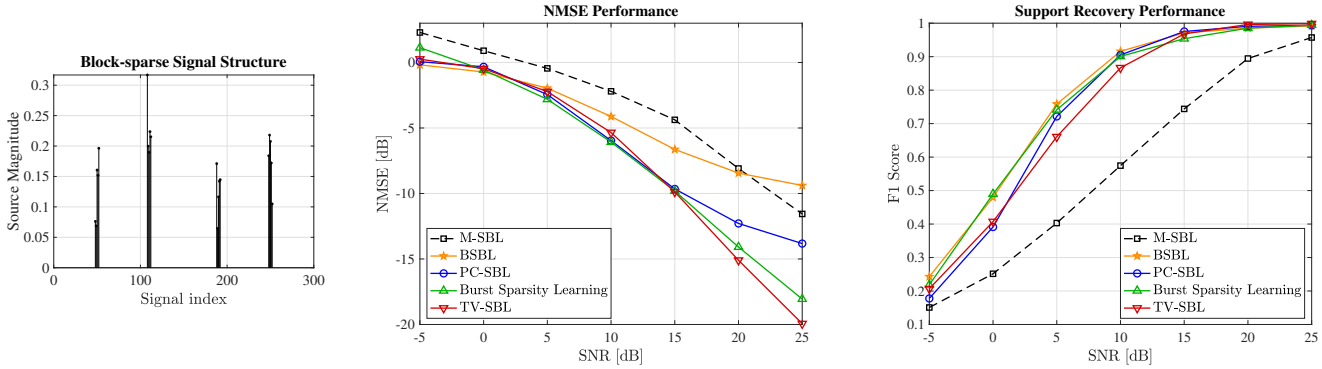
Each entry of noise signal $\mathbf{n}_l$ is generated from $\mathcal{N}(0, \lambda)$ with noise variance λ chosen so that the signal-to-noise ratio (SNR), $10\log_{10}\left(\frac{\mathbb{E}[||\mathbf{A}\mathbf{x}_l||^2]}{\mathbb{E}[||\mathbf{n}_l||^2]}\right)$, varies from -5 to 25 dB. All expectations $\mathbb{E}[\cdot]$ are evaluated over 200 Monte Carlo trials.

We assess the performance with the following two metrics:

1) *Estimation accuracy* is measured by the normalized mean square error (NMSE), $\mathbb{E}\left[\frac{||\hat{\mathbf{X}}-\mathbf{X}||^2}{||\mathbf{X}||^2}\right]$, where $\hat{\mathbf{X}}$ is the estimated source matrix.
2) *Support recovery* is evaluated using the F_1-Score [57], $\mathrm{F}_1 = 2\,\mathbb{E}\left[\frac{\mathrm{precision}\times\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}\right]$, where $\mathrm{precision} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fa}}$, $\mathrm{recall} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{mis}}$, tp is the number of true positives, fa is the number of false alarms, and mis is the number of misdetections. The F_1-Score is a balanced support recovery metric as it penalizes misdetections and false alarms equally. $\mathrm{F}_1 = 1$ implies perfect recovery.

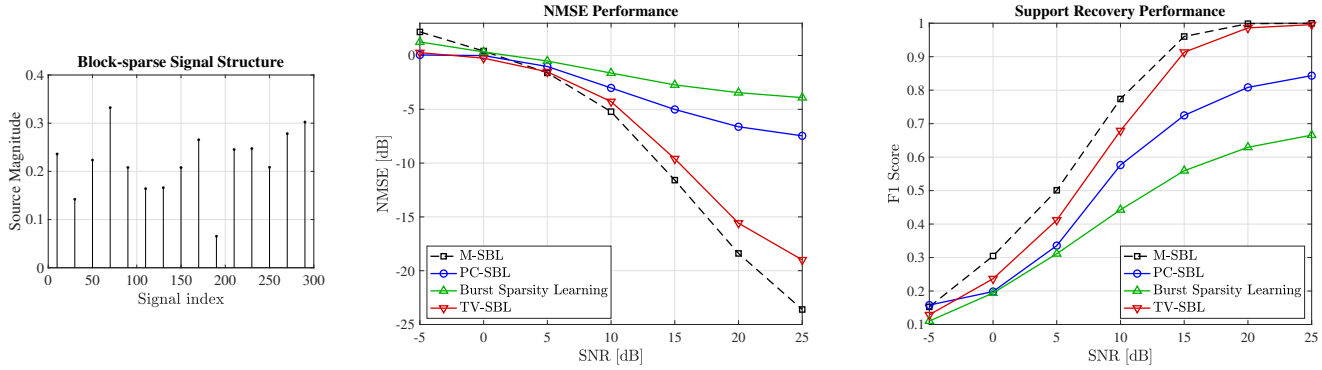
Remark 4. For ease of comparison, we form a support estimate for an algorithm by preserving the K largest rows of $\hat{\mathbf{X}}$ while setting the rest to zero. In practice, the support could also be estimated using a fixed threshold. The choice for setting this

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948



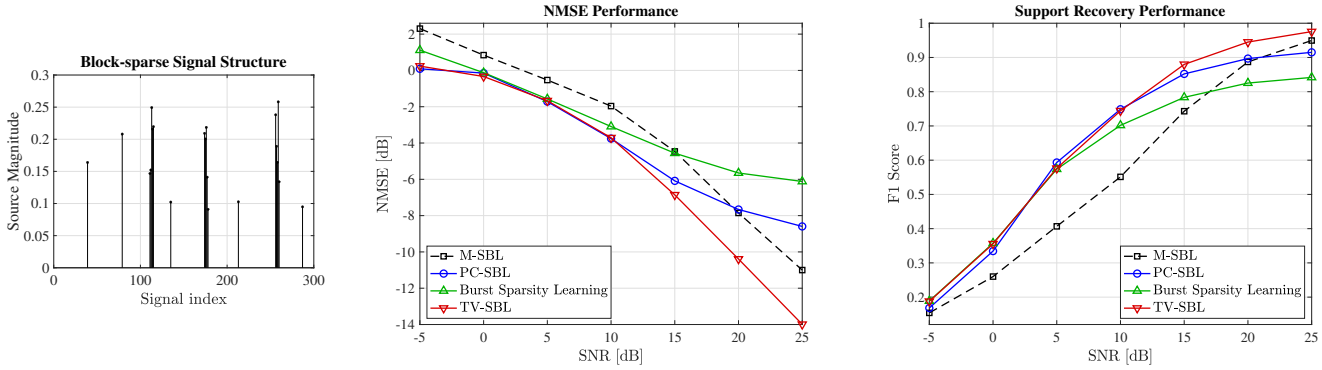
(a) (b) (c)

Fig. 2. Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$: (a) Homogeneous block-sparsity (4 blocks of length 5), (b) NMSE, and (c) Support recovery.



(a) (b) (c)

Fig. 3. Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$: (a) Isolated sparsity (15 blocks of length 1), (b) NMSE, and (c) Support recovery.



(a) (b) (c)

Fig. 4. Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

fixed threshold is not static and we use empirical methods to set the best minimum threshold based on the signal values.

Remark 5. For implementation convenience, we first run the M-SBL algorithm [27] to convergence and use the obtained $\boldsymbol{\gamma}$ values as a warm start (i.e., initialize $\boldsymbol{\gamma}^{(0)}$) for all subsequent block-sparse recovery algorithms.

A. Performance of Different TV Penalties

We first compare the relative performance of the different TV-based SBL regularizers (7), (8), and (11) for the homogeneous block-sparse scenario (see Fig. 1(a)). As evident in Figs. 1(b) and 1(c), all three TV-SBL regularizers outperform the M-SBL algorithm [27] which imposes no prior on $\boldsymbol{\gamma}$. Both conventional TV regularizers (linear TV and log TV) have very similar performance in terms of the NMSE and support recovery[3]. However, the region-aware DoL TV clearly outperforms the conventional TV regularizers. This performance boost reinforces the strategy to selectively penalize signal and zero regions in block-sparse recovery.

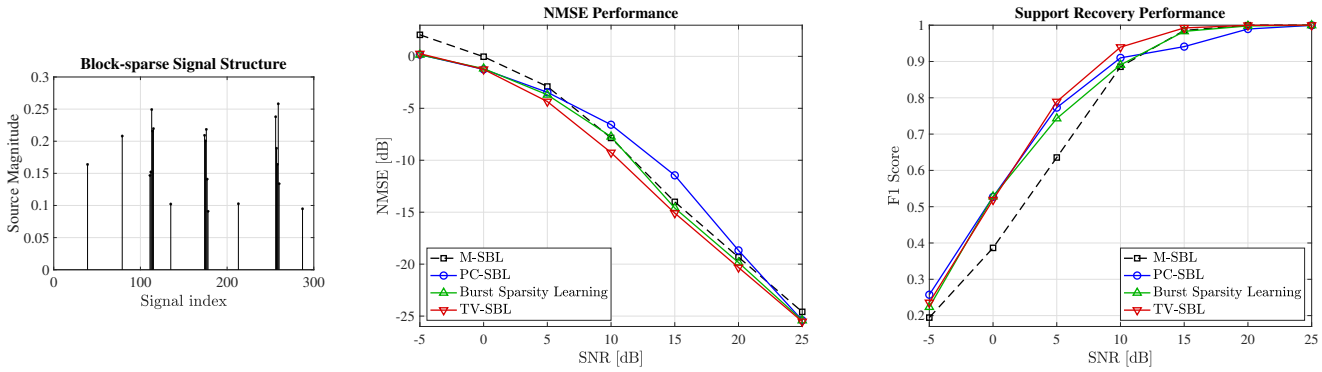
Remark 6. Based on the analysis in Sec. V-A, we use the DoL TV version of the TV-SBL for all subsequent simulations.

B. Comparison with Benchmark Algorithms

We investigate all the three signal classes and compare the performance of our TV-SBL algorithm (using DoL TV) against

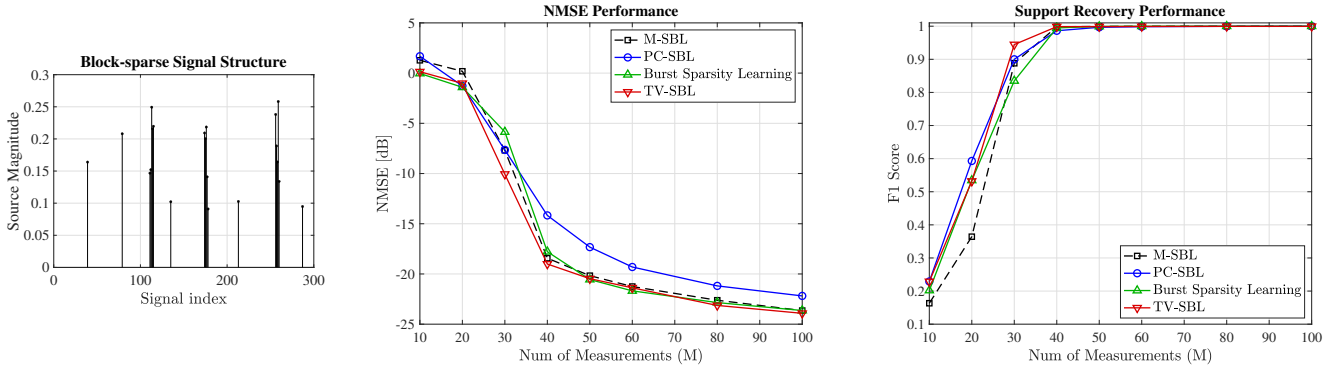
[3]As opposed to the EM-based approaches herein, our results in [1] show that the log TV considerably outperforms the linear TV when optimized via their MM approaches (13) and (15), respectively.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948



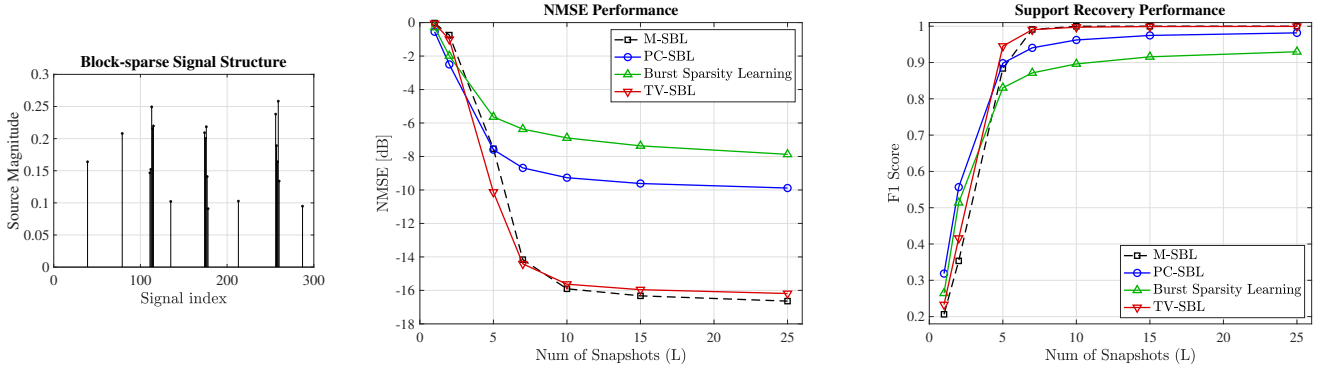
(a) (b) (c)

Fig. 5. Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 45$, and $L = 5$: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.



(a) (b) (c)

Fig. 6. Performance of TV-SBL (DoL TV) versus the benchmark algorithms with varying number of measurements for $N = 300$, $L = 5$ at SNR $= 20$ dB: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.



(a) (b) (c)

Fig. 7. Performance of TV-SBL (DoL TV) versus the benchmark algorithms with varying number of snapshots for $N = 300$, $M = 30$ at SNR $= 20$ dB: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

the following SBL-based block-sparse recovery algorithms: *(i)* BSBL [17], *(ii)* PC-SBL [19], and *(iii)* Burst Sparsity Learning [20]. The M-SBL algorithm [27] is used as a reference to show recovery without regularization.

Remark 7. In order to assess the *robustness* of each algorithm to the changes in block patterns, the parameters of each algorithm were empirically tuned for the homogeneous block-sparse signal, and then left unchanged for random and hybrid sparse signals. This tuning is crucial for all hyperparameter-coupling based algorithms, including our TV-SBL, because it would be evidently optimal to disable the coupling (i.e., set $\beta = 0$ for TV-SBL) for random sparse signals. Thus, we simulate a practical signal recovery scenario where the parameter tuning might not be viable.

1) Homogeneous block-sparse signals: As seen in Fig. 2, all algorithms, unsurprisingly, outperform M-SBL. BSBL, being provided block size and boundary information *apriori*, attains the best F_1-Score (Fig. 2(c)). However, as the NMSE (Fig. 2(b)) illustrates, we need regularization/coupling in addition to fixed block partitioning to recover the signal effectively. TV-SBL shows superior recovery performance at high SNR values; however it is outperformed by Burst Sparsity Learning at lower SNR owing to the softer prior imposed. TV-SBL induces hyperparameter coupling implicitly through the DoL

TV prior (11). This, combined with its region-aware nature, illustrates the utility of TV-SBL for block-sparse recovery.

There is an important caveat to explicit coupling-based priors like PC-SBL and Burst Sparsity Learning: *searching for blocks even when there are none*. We now demonstrate that a softer prior for block-sparsity in TV-SBL enjoys increased flexibility to a block structure, showing TV-SBL's utility beyond homogeneous block-sparsity.

2) Sparse signals: Fig. 3(a) represents the extreme scenario for the block-sparse algorithms: the block size is one. First, as seen in Figs. 3(b) and 3(c), M-SBL achieves the best performance, as expected, establishing a justified performance bound. TV-SBL significantly outperforms the coupling-based algorithms, while being comparable to M-SBL. Explicit hyperparameter coupling in PC-SBL and Burst Sparsity Learning biases the algorithms to block structures, and thus renders them ineffective for isolated sparsity. Using a softer prior, TV-SBL supports block-sparsity without such excessive bias; it is remarkably adept at isolated sparsity as well. This motivates us to analyze a hybrid signal, containing both isolated and block-patterned components, which is typically the case in practice.

3) Hybrid sparse signals: The hybrid block structure in Fig. 4(a) is a representative of a practical scenario for, e.g., multiple-input multiple-output (MIMO) wireless channels. As seen in Figs. 4(b) and 4(c), TV-SBL outperforms all the other algorithms. The soft prior of TV-SBL flexibly accommodates both block-patterned and isolated components, whereas PC-SBL and Burst Sparsity Learning, imposing explicit hyperparameter coupling, are sensitive to isolated components in the underlying signal. This reinforces the fact that TV-SBL presents a *balanced* approach to block-sparse signal recovery; it does not compromise on sparse signal recovery at the cost of block-sparse signal recovery.

C. Effect of Compression Ratio

We highlight the effect of increasing the compression ratio M/N for the hybrid sparse signals in Fig. 5. We increase the number of measurements by $50\ \%$ from that in Fig. 4, i.e., from $M = 30$ to $M = 45$, for the SNR analysis. As observed in Figs. 5(b) and 5(c), the relative gap between the algorithms reduces. Here too, similar to Fig. 4, TV-SBL slightly outperforms the other algorithms. The improved performance for all algorithms is now attributed to the increase in the number of measurements, which reduces the effect of a block-sparsity prior. This is reinforced by analyzing the system performance for varying the number of measurements M at SNR 20 dB, as seen in Fig. 6. As seen from the results in Figs. 6(b) and 6(c), the performance saturates as we increase the number of measurements. This is in line with the general estimation theory in that the impact of a prior is overridden by having increased amount of data to perform the inference. However, it may not always be feasible to acquire more measurements in a fixed data acquisition setup. Nonetheless, for a scenario with a limited number of measurements and snapshots and varying block-sparse structures, our proposed TV-SBL algorithm represents a *robust* regularization-enforced SBL method for *general-patterned block-sparse signal recovery*.

D. Effect of Snapshots

We analyze the performance of the TV-SBL algorithm as we vary the number of snapshots L in Fig. 7. The NMSE (Fig. 7(a)) and support recovery (Fig. 7(b)) are evaluated at SNR 20 dB for the hybrid block-sparse signal structure (Fig. 7(a)). The performance plots illustrate that by increasing the number of snapshots, the performance of M-SBL improves significantly, as is expected from the reduced dependency on the prior. It is interesting to note that while the algorithms using rigid coupling, i.e., PC-SBL and Burst Sparsity Learning saturate at a lower NMSE, the softer prior of the TV-SBL is able to follow the M-SBL algorithm, thereby showing the flexibility of the chosen prior.

VI. Conclusions

We proposed TV-based hyperparameter regularizers for SBL to perform robust block-sparse signal recovery under unknown block patterns. We introduced a new SBL algorithm, TV-SBL, with two perspectives to handle block regularization: 1) the conventional TV regularization (linear TV and log TV) and 2) the RAR using TV (DoL TV). We showed that, after appropriate relaxations, the conventional TV regularization reduces to a sequence of convex optimization problems, enabling a multitude of numerical solvers. The majority of our analysis, algorithms, and experiments focused on the DoL TV regularizer, motivated by its superior performance reported herein. We developed an EM-based alternating optimization solution algorithm, which has universal applicability to all the introduced TV regularizers. The soft TV prior of the TV-SBL – especially when incorporating the region-aware DoL TV – presents a novel balanced perspective on handling block-sparsity. The numerical results show the TV-SBL algorithm to be an efficient method in recovering sparse signals with both block-patterned and isolated components, proving immensely useful for practical signal recovery systems, like mmWave channel estimation with non-uniform sparse scattering.

Appendix

A. Proof of Theorem 1

Using (24) and (25), the right-hand side of (27), $J^{(k)}(\boldsymbol{\gamma}^{(k)})$, is written as

$$J^{(k)}(\boldsymbol{\gamma}^{(k)}) = \sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}). \tag{35}$$

Given an initial value for $J^{(k)}(\boldsymbol{\gamma}^{(k)})$ in (35), by implementing the first step of the algorithm (26), i.e., updating the even elements in $\boldsymbol{\gamma}^{(k+1)}$, we get

$$\begin{aligned} &\textstyle\sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}) \\ &\leq J^{(k)}(\boldsymbol{\gamma}^{(k)}). \end{aligned} \tag{36}$$

The inequality (36) expresses the reduction in the cost function $J^{(k)}(\boldsymbol{\gamma})$ by only minimizing over the even indices of $\boldsymbol{\gamma}$. We now show the same for the odd indices, corresponding to the second step of the algorithm (26), finally leading to (27).

Let us rewrite the left-hand side of (36) equivalently as

$$\begin{aligned} &\textstyle\sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}) = \\ &\textstyle\sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}) + \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i^{(k+1)}). \end{aligned} \tag{37}$$

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

Since the second step of the algorithm (26) minimizes $\sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)})$ over the odd coefficients, by using the equivalence in (37), the second step results in

$$\begin{aligned} J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq & \sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}) \\ & + \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i^{(k+1)}), \end{aligned} \tag{38}$$

where $J^{(k)}(\boldsymbol{\gamma}^{(k+1)})$ represents the M-step cost function value after completing one iteration cycle in the EM algorithm.

Combining (36), (37), and (38) results in $J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq J^{(k)}(\boldsymbol{\gamma}^{(k)})$ in (27), which completes the proof.

B. Segment-wise Optimization for DoL TV Penalty

Using the definition of $f_i^{(k)}(\gamma_i)$ in (22), we rewrite (33) as

$$\begin{aligned} \hat{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, F_1^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + (1+2\beta)\log\gamma_i, \\ \tilde{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, F_0^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + \log\gamma_i, \\ \bar{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, F_2^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + (1-2\beta)\log\gamma_i. \end{aligned} \tag{39}$$

These can be solved by equating the derivative to zero, generating the unconstrained update expressions as

$$\hat{\alpha}_i^{(k)} = \frac{E_i^{(k)}}{1+2\beta}, \quad \tilde{\alpha}_i^{(k)} = E_i^{(k)}, \quad \bar{\alpha}_i^{(k)} = \frac{E_i^{(k)}}{1-2\beta}. \tag{40}$$

We choose the regularization parameter $\beta < 0.5$ to ensure positive minima. Next, we show the validity of the two-step strategy in (33) and (34), separately for each three segments $\hat{\Gamma}_i$, $\bar{\Gamma}_i$, and $\tilde{\Gamma}_i$ given by (30).

Segment $\gamma_i \in \hat{\Gamma}_i$: The derivative of the cost function $F_1^{(k)}(\gamma_i)$ in (39) is

$$\frac{\partial}{\partial\gamma_i} F_1^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + (1+2\beta)\frac{1}{\gamma_i}. \tag{41}$$

It is evident that $\frac{\partial F_1^{(k)}(\gamma_i)}{\partial\gamma_i} > 0$ for $\gamma_i > \hat{\alpha}_i^{(k)}$, where $\hat{\alpha}_i^{(k)}$ is defined in (40). Additionally, if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$, then, by the definition of segment $\hat{\Gamma}_i$ in (30), we must have $\hat{\alpha}_i^{(k)} < \gamma_{i,\max}^{(k)}$. This implies that if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$, then $\hat{\alpha}_i^{(k)} < \gamma_i$, $\forall\gamma_i \in \hat{\Gamma}_i$. Using the fact that (41) is positive for $\gamma_i > \hat{\alpha}_i^{(k)}$, $F_1^{(k)}(\gamma_i)$ is continuously increasing in the segment $\hat{\Gamma}_i$, if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$. Thus, the minimum of $F_1^{(k)}(\gamma_i)$ under the constraint set $\gamma_i \in \hat{\Gamma}_i$ is located either at the stationary point $\hat{\alpha}_i^{(k)}$ (when $\hat{\alpha}_i^{(k)} \in \hat{\Gamma}_i$) or at the lowest value of the segment (when $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$). This leads to the solution of the form (34) as

$$\underset{\gamma_i\in\hat{\Gamma}_i}{\text{argmin}}\; F_1^{(k)}(\gamma_i) = \max\big(\gamma_{i,\max}^{(k)},\, \hat{\alpha}_i^{(k)}\big). \tag{42}$$

Segment $\gamma_i \in \bar{\Gamma}_i$: The derivative of the cost function $F_2^{(k)}(\gamma_i)$ in (39) is

$$\frac{\partial}{\partial\gamma_i} F_2^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + (1-2\beta)\frac{1}{\gamma_i}. \tag{43}$$

It is evident that $\frac{\partial F_2^{(k)}(\gamma_i)}{\partial\gamma_i} < 0$ for $0 < \gamma_i < \bar{\alpha}_i^{(k)}$, where $\bar{\alpha}_i^{(k)}$ is defined in (40). By the definition of segment $\bar{\Gamma}_i$ in (30), if $\bar{\alpha}_i^{(k)} \notin \bar{\Gamma}_i$, then $\bar{\alpha}_i^{(k)} > \gamma_{i,\min}^{(k)}$; this implies $\bar{\alpha}_i^{(k)} > \gamma_i$, $\forall\gamma_i \in \bar{\Gamma}_i$. Using the fact that (43) is negative for $0 < \gamma_i < \bar{\alpha}_i^{(k)}$, $F_2^{(k)}(\gamma_i)$ is continuously decreasing in the segment $\bar{\Gamma}_i$ if $\bar{\alpha}_i^{(k)} \notin \bar{\Gamma}_i$. Following the logic similar to segment $\hat{\Gamma}_i$, the solution is of the form (34), i.e.,

$$\underset{\gamma_i\in\bar{\Gamma}_i}{\text{argmin}}\; F_2^{(k)}(\gamma_i) = \max\big(0,\, \min\big\{\gamma_{i,\min}^{(k)},\, \bar{\alpha}_i^{(k)}\big\}\big) \tag{44}$$

Segment $\gamma_i \in \tilde{\Gamma}_i$: The derivative of the cost function $F_0^{(k)}(\gamma_i)$ in (39) is

$$\frac{\partial}{\partial\gamma_i} F_0^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + \frac{1}{\gamma_i}. \tag{45}$$

It is evident that *(i)* $\frac{\partial F_0^{(k)}(\gamma_i)}{\partial\gamma_i} < 0$ for $0 < \gamma_i < \tilde{\alpha}_i^{(k)}$, and *(ii)* $\frac{\partial F_0^{(k)}(\gamma_i)}{\partial\gamma_i} > 0$ for $\gamma_i > \tilde{\alpha}_i^{(k)}$, where $\tilde{\alpha}_i^{(k)}$ is defined in (40). By the definition of segment $\tilde{\Gamma}_i$ in (30), if $\tilde{\alpha}_i^{(k)} \notin \tilde{\Gamma}_i$, then either $\tilde{\alpha}_i^{(k)} < \gamma_{i,\min}^{(k)}$ or $\tilde{\alpha}_i^{(k)} > \gamma_{i,\max}^{(k)}$. If $\tilde{\alpha}_i^{(k)} < \gamma_{i,\min}^{(k)}$, then $F_0^{(k)}(\gamma_i)$ is increasing in $\tilde{\Gamma}_i$; in this case, $\tilde{\gamma}_i^{(k)} = \gamma_{i,\min}^{(k)}$. Analogously, if $\tilde{\alpha}_i^{(k)} > \gamma_{i,\max}^{(k)}$, then $\tilde{\gamma}_i^{(k)} = \gamma_{i,\max}^{(k)}$. Putting these conditions together, we obtain the solution in (34) as

$$\underset{\gamma_i\in\tilde{\Gamma}_i}{\text{argmin}}\; F_0^{(k)}(\gamma_i) = \min\Big(\gamma_{i,\max}^{(k)},\, \max\Big\{\gamma_{i,\min}^{(k)},\, \tilde{\alpha}_i^{(k)}\Big\}\Big). \tag{46}$$

C. Unconstrained Updates in (33) *for Linear TV*

We begin by reformulating the M-step (23) to correspond to the linear TV penalty in (7) as

$$J^{(k)}(\boldsymbol{\gamma}) = \sum_{i=1}^N \left[\frac{E_i^{(k)}}{\gamma_i} + \log\gamma_i\right] + \beta\sum_{i=2}^N |\gamma_i - \gamma_{i-1}|. \tag{47}$$

We majorize $\log\gamma_i$ by its first-order Taylor approximation at point $\gamma_i^{(k)}$, $i = 1,\ldots,N$. Consequently, (47) is majorized as

$$\hat{J}^{(k)}(\boldsymbol{\gamma}) = \sum_{i=1}^N \left[\frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i\right] + \beta\sum_{i=2}^N |\gamma_i - \gamma_{i-1}|, \tag{48}$$

where $q_i^{(k)} = \frac{1}{\gamma_i^{(k)}+\epsilon}$ and ϵ is a small stability parameter. Following the steps similar to Sec. IV-A and IV-B, we obtain the majorized cost function equivalent to (29) as

$$\begin{aligned} \hat{F}_i^{(k)}(\gamma_i;\gamma_{i,\max}^{(k)},\gamma_{i,\min}^{(k)}) &= \frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i \\ &+\beta\,\big(\big|\gamma_i - \gamma_{i,\max}^{(k)}\big| + \big|\gamma_i - \gamma_{i,\min}^{(k)}\big|\,\big), \end{aligned} \tag{49}$$

Accordingly, using the definition of $f_i^{(k)}(\gamma_i)$ in (22), we write the updates in (33) for linear TV as

$$\begin{aligned} \hat{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, \tfrac{E_i^{(k)}}{\gamma_i} + (q_i^{(k)}+2\beta)\gamma_i \\ \tilde{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, \tfrac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i \\ \bar{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\, \tfrac{E_i^{(k)}}{\gamma_i} + (q_i^{(k)}-2\beta)\gamma_i, \end{aligned} \tag{50}$$

For the first two problems in (50), differentiating with respect to γ_i and equating to zero, we get $\hat{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+2\beta}}$ and $\tilde{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)}}}$. For the third problem in (50), it is possible that $q_i^{(k)} - 2\beta < 0$, i.e., the cost function is unbounded below and theoretically, $\bar{\alpha}_i^{(k)} = \infty$. In a practical implementation, we account for the negativity $q_i^{(k)} - 2\beta < 0$ via setting a large value

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3144948

$\bar{\alpha}_i^{(k)}$. Thus, the update rule becomes $\bar{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max(\epsilon_0, q_i^{(k)} - 2\beta)}}$, where ϵ_0 is a small number.

D. Unconstrained Updates in (33) for Log TV

We begin by formulating the M-step (23) with respect to the log TV penalty in (8) as

$$J^{(k)}(\boldsymbol{\gamma}) = \sum_{i=1}^{N} \left[\frac{E_i^{(k)}}{\gamma_i} + \log \gamma_i \right] + \beta \sum_{i=2}^{N} \log(|\gamma_i - \gamma_{i-1}| + \epsilon_1). \tag{51}$$

Besides majorizing $\log \gamma_i$ similar to the linear TV case in Appendix C, we majorize $\log(|\gamma_i - \gamma_{i-1}| + \epsilon_1)$ by its first-order Taylor approximation at point $(\gamma_i^{(k)} - \gamma_{i-1}^{(k)})$, $i = 2, \ldots, N$. Consequently, (51) is majorized as

$$\tilde{J}^{(k)}(\boldsymbol{\gamma}) = \sum_{i=1}^{N} \left[\frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)} \gamma_i \right] + \beta \sum_{i=2}^{N} c_i^{(k)} |\gamma_i - \gamma_{i-1}|, \tag{52}$$

where $q_i^{(k)} = \frac{1}{\gamma_i^{(k)} + \epsilon}$ and $c_i^{(k)} = \frac{1}{|\gamma_i^{(k)} - \gamma_{i-1}^{(k)}| + \epsilon_1}$.

Following the steps similar to Sec. IV-A and IV-B, we obtain the majorized cost function having the form similar to (29):

$$\begin{aligned} \tilde{F}_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}) &= \frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)} \gamma_i \\ &+ \beta \left(a_i^{(k)} \left| \gamma_i - \gamma_{i,\max}^{(k)} \right| + b_i^{(k)} \left| \gamma_i - \gamma_{i,\min}^{(k)} \right| \right), \end{aligned} \tag{53}$$

where $a_i^{(k)} = \frac{1}{|\gamma_i^{(k)} - \gamma_{\max}^{(k)}| + \epsilon_1}$ and $b_i^{(k)} = \frac{1}{|\gamma_i^{(k)} - \gamma_{\min}^{(k)}| + \epsilon_1}$. Observing the similarity to (49), the unconstrained updates (33) for log TV become

$$\begin{aligned} \hat{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \left(q_i^{(k)} + \beta(a_i^{(k)} + b_i^{(k)}) \right) \gamma_i \\ \tilde{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \left(q_i^{(k)} + \beta(b_i^{(k)} - a_i^{(k)}) \right) \gamma_i \\ \bar{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \left(q_i^{(k)} - \beta(a_i^{(k)} + b_i^{(k)}) \right) \gamma_i. \end{aligned} \tag{54}$$

Based on the same logic as for the linear TV updates, the solutions to these unconstrained updates are $\hat{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)} + \beta(a_i^{(k)} + b_i^{(k)})}}$, $\tilde{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)} + \beta(b_i^{(k)} - a_i^{(k)})\right)}}$, and $\bar{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)} - \beta(a_i^{(k)} + b_i^{(k)})\right)}}$.

REFERENCES

[1] A. Sant, M. Leinonen, and B. Rao, "General total variation regularized sparse Bayesian learning for robust block-sparse signal recovery," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Canada, Jun. 6–11 2021, pp. 5604–5608.

[2] E. J. Candés, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] S. Hur, S. Baek, B. Kim, Y. Chang, A. F. Molisch, T. S. Rappaport, K. Haneda, and J. Park, "Proposal on millimeter-wave channel modeling for 5G cellular system," *IEEE J. Select. Topics Signal Process.*, vol. 10, no. 3, pp. 454–469, April 2016.

[5] C. Gustafson, K. Haneda, S. Wyne, and F. Tufvesson, "On mm-wave multipath clustering and channel modeling," *IEEE Trans. Antennas Propagat.*, vol. 62, no. 3, pp. 1445–1455, March 2014.

[6] K. Haneda, "Channel models and beamforming at millimeter-wave frequency bands," *IEICE Transactions*, vol. 98-B, pp. 755–772, 2015.

[7] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[8] E. Van Den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2009.

[9] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[10] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, 2010.

[11] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, 2009.

[12] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity." *Journal of Machine Learning Research*, vol. 12, no. 11, 2011.

[13] T. Peleg, Y. C. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2286–2303, May 2012.

[14] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, p. 211244, Sep. 2001. [Online]. Available: https://doi.org/10.1162/15324430152748236

[15] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[16] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Select. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.

[17] ——, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 2009–2015, April 2013.

[18] L. Yu, H. Sun, J.-P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal processing*, vol. 92, no. 1, pp. 259–269, 2012.

[19] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, Jan 2015.

[20] J. Dai, A. Liu, and H. C. So, "Non-uniform burst-sparsity learning for massive MIMO channel estimation," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 1075–1087, Feb 2019.

[21] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, November 2008.

[22] L. Wang, L. Zhao, S. Rahardja, and G. Bi, "Alternative to extended block sparse Bayesian learning and its relation to pattern-coupled sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2759–2771, May 2018.

[23] M. Shekaramiz, T. K. Moon, and J. H. Gunther, "Bayesian compressive sensing of sparse signals with unknown clustering patterns," *Entropy*, vol. 21, no. 3, 2019.

[24] L. Wang, L. Zhao, L. Yu, J. Wang, and G. Bi, "Structured Bayesian learning for recovery of clustered sparse signal," *Signal Processing*, vol. 166, p. 107255, 2020.

[25] R. M. Neal, *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag, 1996.

[26] D. J. MacKay and C. Laboratory, "Bayesian non-linear modelling for the prediction competition," in *In ASHRAE Transactions, V.100, Pt.2*. ASHRAE, 1994, pp. 1053–1062.

[27] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, 2007.

[28] P. Pal and P. P. Vaidyanathan, "Pushing the limits of sparse support recovery using correlation information," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 711–726, 2015.

[29] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems*, 2002, pp. 383–389.

[30] R. R. Pote and B. D. Rao, "Robustness of sparse Bayesian learning in correlated environments," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Barcelona, Spain, May 4-8, 2020, pp. 9100–9104.

[31] M. Al-Shoukairi, P. Schniter, and B. D. Rao, "A GAMP-based low complexity sparse Bayesian learning algorithm," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 294–308, 2018.

[32] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Select. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, 2010.

[33] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational em algorithms for non-gaussian latent variable models," *Advances in neural information processing systems*, vol. 18, p. 1059, 2006.

[34] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 53–63, 2010.

[35] D. Wipf, J. Palmer, B. Rao, and K. Kreutz-Delgado, "Performance evaluation of latent variable models with sparse priors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, 2007, pp. II–453–II–456.

[36] T. A. Srikrishnan and B. D. Rao, "Addressing the noise variance problem in sparse Bayesian learning," in *Proc. Asilomar Conf. Signals, Syst., Comp.*, 2018, pp. 1974–1979.

[37] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005. [Online]. Available: https://doi.org/10.1137/040605412

[38] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727899290242F

[39] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 813–824, 1998.

[40] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen, "Image restoration using total variation with overlapping group sparsity," *Information Sciences*, vol. 295, pp. 232–246, 2015.

[41] J. Huang and F. Yang, "Compressed magnetic resonance imaging based on wavelet sparsity and nonlocal total variation," in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2012, pp. 968–971.

[42] X. Zhang, T. Bai, H. Meng, and J. Chen, "Compressive sensing-based ISAR imaging via the combination of the sparsity and nonlocal total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 5, pp. 990–994, 2014.

[43] M. Eickenberg, E. Dohmatob, B. Thirion, and G. Varoquaux, "Grouping total variation and sparsity: Statistical learning with segmenting penalties," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 685–693.

[44] H. K. Aggarwal and A. Majumdar, "Hyperspectral unmixing in the presence of mixed noise using joint-sparsity and total variation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4257–4266, 2016.

[45] W. He, H. Zhang, and L. Zhang, "Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sensing*, vol. 55, no. 7, pp. 3909–3921, 2017.

[46] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal $\ell 1$-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 6, pp. 797–829, 2006.

[47] E. J. Candés, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[48] Y. Shen, J. Fang, and H. Li, "Exact reconstruction analysis of log-sum minimization for compressed sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1223–1226, 2013.

[49] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in Neural Information Processing Systems*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1625–1632.

[50] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[51] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[52] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[53] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[54] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[55] J. Bezdek, R. Hathaway, R. Howard, C. Wilson, and M. Windham, "Local convergence analysis of a grouped variable version of coordinate descent," *Journal of Optimization Theory and Applications*, vol. 54, no. 3, pp. 471–477, 1987.

[56] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[57] N. Chinchor, "MUC-4 evaluation metrics," in *In Proceedings of the Fourth Message Understanding Conference*, 1992, pp. 22–29.