

Adaptive and Self-Tuning SBL With Total Variation Priors for Block-Sparse Signal Recovery

Hamza Djelouat , Reijo Leinonen , Mikko J. Sillanpää , Bhaskar D. Rao , *Life Fellow, IEEE*,
and Markku Juntti , *Fellow, IEEE*

Abstract—This letter addresses the problem of estimating block sparse signal with unknown group partitions in a multiple measurement vector (MMV) setup. We propose a Bayesian framework by applying an adaptive total variation (TV) penalty on the hyper-parameter space of the sparse signal. The main contributions are two-fold. 1) We extend the TV penalty beyond the immediate neighbor, thus enabling better capture of the signal structure. 2) A dynamic framework is provided to learn the regularization weights for the TV penalty based on the statistical dependencies between the entries of tentative blocks, thus eliminating the need for fine-tuning. The superior performance of the proposed method is empirically demonstrated by extensive computer simulations with the state-of-art benchmarks. The proposed solution exhibits both excellent performance and robustness against sparsity model mismatch.

Index Terms—Compressive sensing, block-sparsity, total variation, SBL, ADMM.

I. INTRODUCTION

BLOCK sparsity, where nonzero elements occur in clusters rather than uniformly in the data, provides a powerful framework for signal recovery. Such a grouping pattern arises in many practical scenarios, such as near-field and millimeter-wave channel estimation [1], [2], and user activity detection in machine-type communications [3]. While block sparsity provides valuable prior knowledge about the signal structure, leveraging this information presents a key challenge. The recovery process must simultaneously determine the block locations, their extent, and their values, making it substantially more complex than conventional sparse signal recovery where only individual nonzero elements need to be identified [4].

Sparse recovery algorithms developed under Compressive sensing (CS) [5], [6] framework excel at promoting sparsity but ignore the inherent block structures present in many signals. For scenarios where block boundaries are known a priori, the CS algorithms have been adapted to exploit this structure.

Received 17 January 2025; revised 21 March 2025; accepted 22 March 2025. Date of publication 1 April 2025; date of current version 16 April 2025. This work was supported in part by the Research Council of Finland through 6GWiCE Project under Grant 319485 and in part by 6G Flagship Program under Grant 369116. The work of Bhaskar D. Rao was supported by the NSF under Grant 2225617. The associate editor coordinating the review of this article and approving it for publication was Dr. Lu Gan. (*Corresponding author: Hamza Djelouat.*)

Hamza Djelouat and Markku Juntti are with the Centre for Wireless Communications – Radio Technologies, University of Oulu, FI-90014 Oulu, Finland (e-mail: hamzadjelouat@oulu.fi).

Reijo Leinonen and Mikko J. Sillanpää are with the Research Unit of Mathematical Sciences, University of Oulu, FI-90014 Oulu, Finland.

Bhaskar D. Rao is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla CA 92093 USA.

Digital Object Identifier 10.1109/LSP.2025.3556790

Those include block compressive sampling matching pursuit (block-CoSaMP) [7], block orthogonal matching pursuit (block-OMP) [8], and block-sparse Bayesian learning (BSBL) [9]. By incorporating signal-specific block structures, these recovery algorithms achieve significantly better performance than their conventional counterparts.

The practical scenarios present an additional challenge of the *unknown block boundaries and their sizes*. This has spurred development of adaptive techniques that simultaneously promote block sparsity and detect boundaries, primarily through modeling inter-element dependencies within the signal.

Sparse signal recovery with unknown boundaries has been approached through two main frameworks: Bayesian and regularization-based methods. For instance, pattern-coupled (PC) sparse Bayesian learning (SBL) and its variants [10], [11], [12], [13] promote block sparsity by hard-coupling neighboring elements in the signal's variance vector. Meanwhile, regularization approaches such as fused least absolute shrinkage and selection operator (LASSO) [14] employs total variation (TV) penalty to discourage rapid signal fluctuations. A recent work [15] combined these two approaches by applying the TV regularization to the signal hyper-parameters within an SBL framework, effectively reducing transitions between zero and non-zero regions. However, all of these frameworks have two key limitations: they restrict smoothing to the immediate neighbor only, overlooking broader structural patterns, and require careful tuning of the block-sparsity promoting term – a challenge when block sparsity varies [15].

This letter introduces a novel solution to block-sparse signal recovery by formulating the problem using an SBL cost function augmented by a *log-based TV* penalty with *learnable* weights. Specifically, the proposed approach aims to (i) extend the hyper-parameter coupling from a single neighbor to multiple consecutive neighbors, thereby enhancing the model's ability to capture local structures; and (ii) to provide a framework to dynamically and automatically learn the regularization parameter based on the signal statistics on each iteration. Finally, to ensure computational efficiency, we employ an alternating direction method of multipliers (ADMM) framework [16], which allows for closed-form updates at each step of the optimization process. We validate the effectiveness of the proposed approach via extensive experiments, showing significant improvements in signal recovery performance over existing state-of-the-art methods in terms of quality and robustness to model mismatch.

II. BLOCK-SPARSE RECOVERY PROBLEM OVERVIEW

A. System Model

We consider the estimation of the unknown block-sparse signal ensemble $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M] \in \mathbb{C}^{N \times M}$ from a set of noisy

linear measurements $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_M] \in \mathbb{C}^{L \times M}$ as

$$\mathbf{y}_m = \mathbf{A}\mathbf{x}_m + \mathbf{w}_m, \quad m = 1, \dots, M, \quad (1)$$

where $\mathbf{A} \in \mathbb{C}^{L \times N}$ is a fixed and known measurement matrix, and each $\mathbf{w}_m \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_L)$ is the additive noise vector.

We focus on a signal ensemble that exhibits two key properties: (i) a *row-sparse* structure, meaning all signals $\mathbf{x}_m, \forall m$, share the same support, and (ii) a *block-sparse* structure, where each signal $\mathbf{x}_m, \forall m$, comprises an *unknown* number of blocks with *varying* sizes. The unknown signal \mathbf{x}_m is modeled to follow a multivariate complex Gaussian distribution as

$$\mathbf{x}_m \sim \mathcal{CN}(\mathbf{0}, \text{diag}(\boldsymbol{\gamma})), \quad m = 1, \dots, M \quad (2)$$

where $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_N]^T \in \mathbb{R}_+^N$ represents the *unknown hyper-parameter vector* controlling the sparsity of each \mathbf{x}_m .

B. SBL Framework

Given the system model in (1), the likelihood function is Gaussian or $p(\mathbf{y}_m|\mathbf{x}_m; \sigma) = \mathcal{CN}(\mathbf{A}\mathbf{x}_m, \sigma^2)$. Thus by using the conjugate Gaussian prior in (2) along the likelihood of Gaussian density, the posterior density $p(\mathbf{x}_m|\mathbf{y}_m; \boldsymbol{\gamma})$ is also Gaussian, given as

$$p(\mathbf{x}_m|\mathbf{y}_m; \boldsymbol{\gamma}) \sim \mathcal{CN}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_\mathbf{x}), \quad (3)$$

where

$$\boldsymbol{\mu}_m = \frac{1}{\sigma^2} \boldsymbol{\Sigma}_\mathbf{x} \mathbf{A}^H \mathbf{y}_m, \quad \boldsymbol{\Sigma}_\mathbf{x} = \left(\frac{1}{\sigma^2} \mathbf{A}^H \mathbf{A} + \text{diag}(\boldsymbol{\gamma})^{-1} \right)^{-1}, \quad (4)$$

For known $\boldsymbol{\gamma}$, the estimated signal is given as $\mathbf{X} = [\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M]$. The SBL framework aims to estimate the hyper-parameter vector $\boldsymbol{\gamma}$ via Type-II maximum a posteriori (MAP) estimation as

$$\begin{aligned} \hat{\boldsymbol{\gamma}} &= \underset{\boldsymbol{\gamma} \geq 0}{\text{argmax}} \log p(\boldsymbol{\gamma}|\mathbf{Y}) \\ &= \underset{\boldsymbol{\gamma} \geq 0}{\text{argmin}} \log \det(\boldsymbol{\Sigma}_\mathbf{y}) + \text{Tr}[\mathbf{Y} \boldsymbol{\Sigma}_\mathbf{y}^{-1}] - \log p(\boldsymbol{\gamma}), \end{aligned} \quad (5)$$

where $\boldsymbol{\Sigma}_\mathbf{y} = \sigma^2 \mathbf{I} + \mathbf{A} \text{diag}(\boldsymbol{\gamma}) \mathbf{A}^H$ is the measurement model covariance matrix and $\log p(\boldsymbol{\gamma})$ is the *hyper-prior* on $\boldsymbol{\gamma}$. The problem (5) is non-convex due to the logdet term. Therefore, several approaches have been proposed to approximate the solution such via block coordinate descent [17], fixed point iteration [18], and expectation maximization (EM) [19], [20]. We will employ the latter to solve the problem.

The EM-SBL iterates between two steps as follows

E-step: The joint distribution is given as

$$p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\gamma}) = p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}). \quad (6)$$

At any particular iteration (k), the E-step with respect to $\boldsymbol{\gamma}$ is evaluated by averaging out the hidden variable \mathbf{X} as

$$\begin{aligned} Q(\boldsymbol{\gamma}^k | \boldsymbol{\gamma}^{(k-1)}) &= \mathbb{E}_{p(\mathbf{X}|\mathbf{Y}, \boldsymbol{\gamma}^{(k-1)})} \left[\log p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}) \right] \\ &\stackrel{(a)}{\propto} \mathbb{E}_{p(\mathbf{X}|\mathbf{Y}, \boldsymbol{\gamma}^{(k-1)})} \left[\log p(\mathbf{X}|\boldsymbol{\gamma}) + \log p(\boldsymbol{\gamma}) \right] \\ &= - \sum_{i=1}^N M \left[\log(\gamma_i) + \frac{s_i^{(k-1)}}{\gamma_i} \right] + \log p(\boldsymbol{\gamma}), \end{aligned} \quad (7)$$

where (a) is obtained by dropping terms that don't depend on $\boldsymbol{\gamma}$ and $s_i^{(k-1)} = \sum_{i,i}^{(k-1)} + \frac{1}{M} \|\boldsymbol{\mu}_i^{(k-1)}\|^2$.

M-step: The hyper-parameter is updated as follows

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\text{argmin}} -Q(\boldsymbol{\gamma}^k | \boldsymbol{\gamma}^{(k-1)}) = \underset{\boldsymbol{\gamma}}{\text{argmin}} f(\boldsymbol{\gamma}) - \frac{1}{M} \log p(\boldsymbol{\gamma}), \quad (8)$$

where

$$f(\boldsymbol{\gamma}) = \sum_{i=1}^N f(\gamma_i) = \sum_{i=1}^N \log(\gamma_i) + \frac{s_i}{\gamma_i}. \quad (9)$$

To promote block-sparse solutions in the EM-SBL framework, it is essential to employ **prior functions** that simultaneously encourage sparsity and capture the inherent block structures in the signal. In the existing literature, block sparsity is typically enforced through two main approaches: 1) a hard prior coupling scheme where each hyper-parameter is defined as $\gamma_i \triangleq f(\gamma_i, \gamma_{i+1}, \gamma_{i-1}, \beta)$, as seen in PC-SBL inspired solutions, and 2) a soft prior formulation given by $-\log p(\boldsymbol{\gamma}) = \beta \sum_{i=2}^N |\log(\gamma_i) - \log(\gamma_{i-1})|$, where $\beta > 0$ is a regularization parameter controlling the coupling strength in both approaches. However, these approaches suffer from two major limitations: they only consider immediate neighboring elements in their coupling mechanisms, and they require careful tuning of the parameter β , making them susceptible to sparsity model mismatches, particularly when isolated non-zero elements appear in the signal.

We aim to improve the existing solutions and to bridge some gaps to overcome the limitations of the proposed approaches in terms of accuracy and robustness of the estimation. We will first discuss the design of the prior function, followed by the detailed derivation of the solution for the M-step.

III. PROPOSED SOLUTION

Before proceeding to solve problem (8), we first discuss the appropriate choice of $p(\boldsymbol{\gamma})$ to promote a *robust structured* sparse solution. Our approach builds upon the framework introduced in [15] and enhances the SBL inference framework through the following key ideas. 1) **Neighborhood smoothing** extends the hyper-parameter coupling from a single neighbor to multiple consecutive neighbors, hence, improving the model's ability to adapt to the signal structure and capture local dependencies between the elements of the signal. 2) **Automatic regularization tuning** by learning the set of regularization parameters $\{\beta_{i,j}\}$, where each $\beta_{i,j}$ is unique for each pair of γ_i and γ_j . This is achieved by adaptively updating $\{\beta_{i,j}\}$ at each EM iteration based on the signal statistics as we will show later.

A. Design of Hyper-Prior $p(\boldsymbol{\gamma})$

To this end, we propose the following hyper-prior function:

$$-\log p(\boldsymbol{\gamma}) = \sum_{i=2}^N \sum_{j=1}^{i-1} \beta_{i,j} |g(\gamma_i) - g(\gamma_j)|, \quad (10)$$

where $\beta_{i,j} \in [0, 1]$ is a weight parameter, and $g(\cdot)$ is a function that measures the number of transitions between zero and non-zero regions. Ideally, we would choose $g(\cdot) = \mathbb{I}(\cdot)$ (the indicator function), but this would result in a binary optimization problem that is computationally intractable. Instead, we use the approximation $g(\cdot) = \log(\cdot)$, as the log function provides a suitable approximation to the counting function $\mathbb{I}(\cdot)$ [21]. This

letter defines $\{\beta_{i,j}\}$ for each pair of $\{\gamma_i, \gamma_j\}$ as

$$\beta_{i,j} = \frac{1}{Z_i} \psi[d(\log(\gamma_i), \log(\gamma_j))], \quad (11)$$

where Z_i is a normalization factor that ensures that the weights sum to 1, $\psi[\cdot]$ is a kernel decay function and $d(\cdot)$ is a distance function to measure the similarity between the sparsity profiles of the i th and j th elements. Both $\psi(\cdot)$ and $d(\cdot)$ are designed to enforce a zero penalty on the smoothing between i th and j th elements if their sparsity profile is different, while setting a positive weight when the i th and the j th elements have a similar sparsity profile. To this end, we set ψ as a negative exponential and set $d(\cdot)$ as an Euclidean norm. Subsequently, we define $\beta_{i,j}$ at each EM iteration (k) as follows

$$\beta_{i,j}^{(k)} = \begin{cases} \exp(-\|\log(\gamma_i^{(k-1)}) - \log(\gamma_j^{(k-1)})\|^2), & \text{if } j \in \Omega_i \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

where Ω_i is a set of the closest neighbors of the i th element to be selected empirically.

B. ADMM Solution for the M-Step

We use the proposed $p(\gamma)$ and express the optimization problem for the M-step at the k th EM iteration as¹

$$\hat{\gamma} = \arg \min_{\gamma} f(\gamma) + \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^{i-1} \beta_{i,j} |\log(\gamma_i) - \log(\gamma_j)|. \quad (13)$$

Given the non-convexity of (13) and the parameter coupling introduced by the TV term, efficient computation presents a significant challenge. To address this, we employ an ADMM framework [16] for solving the M-step (13), which enables sequential closed-form updates and yields a more computationally efficient solution as detailed below.

First, we introduce the auxiliary variable $\mathbf{C} \in \mathbb{R}^{N \times N}$ to transform (13) into a bi-convex problem as follows

$$\begin{aligned} \arg \min_{\gamma, \mathbf{C}} f(\gamma) + \frac{1}{M} \sum_{i=2}^N \sum_{j=1}^{i-1} \beta_{i,j} |C_{i,j}| \\ \text{s.t. } C_{i,j} = (\log(\gamma_i) - \log(\gamma_j)), \forall i, j \in \mathcal{N}. \end{aligned} \quad (14)$$

The objective function in (14) provides an excellent framework to apply an alternating optimization technique such as ADMM. Subsequently, we write the augmented Lagrangian function as

$$\mathcal{L}(\gamma, \mathbf{C}) = f(\gamma) + \sum_{i=2}^N \sum_{j=1}^{i-1} \left[\frac{\beta_{i,j}}{M} |C_{i,j}| + \frac{\rho}{2} \left\| C_{i,j} - \bar{\gamma}_{i,j} + \frac{\lambda_{i,j}}{\rho} \right\|^2 \right], \quad (15)$$

where $\bar{\gamma}_{i,j} = \log(\gamma_i) - \log(\gamma_j)$, $\lambda \in \mathbb{R}^{N \times N}$ denotes the dual variable matrix, and ρ is a real small positive variable.

The ADMM framework solves (13) by iteratively minimizing $\mathcal{L}(\cdot)$ over the primal variables (γ, \mathbf{C}) , followed by updates to the dual variables λ [16]. We detail the derivation of each ADMM iteration step (t) below.

Algorithm 1: The Proposed Algorithm.

Input: $\mathbf{A}, \mathbf{Y}, \epsilon, \rho, t_{\max}, k_{\max}, \{\Omega_i\}_{i=1}^N$.
Initialization: $\gamma^{(0)}, \mathbf{C}^{(0)}, \lambda^{(0)}, t = 1, k = 1$.
1 while $k < k_{\max}$ **or** $\|\mu_x^{(k)} - \mu_x^{(k+1)}\|^2 > \epsilon$ **do**
2 $\Sigma_x^{(k+1)} = \left(\frac{1}{\sigma^2} \mathbf{A}^H \mathbf{A} + \text{diag}(\gamma^{(k)-1}) \right)^{-1}$
3 $\mu_x^{(k+1)} = \frac{1}{\sigma^2} \Sigma_x^{(k+1)} \mathbf{A}^H \mathbf{y}$
4 $\beta_{i,j} = \exp(-\|\log(\gamma_i^{(k)}) - \log(\gamma_j^{(k)})\|^2), \forall i, \forall j$
5 **while** $t < t_{\max}$ **or** $\|\gamma^{(t)} - \gamma^{(t+1)}\|^2 > \epsilon$ **do**
6 Update $\gamma^{(t+1)}$ using (17)
7 Update $\mathbf{C}^{(t+1)}$ using (19)
8 Update $\lambda^{(t+1)}$ using (20)
9 $t \leftarrow t + 1$
10 $k \leftarrow k + 1$

1) γ -Update: First, we update each element of the hyper-parameter vector sequentially as

$$\begin{aligned} \gamma_i^{(t+1)} = \arg \min_{\gamma_i} \log(\gamma_i) + \frac{\rho}{2} \sum_{\substack{j \in \Omega_i \\ j < i}} \|\bar{a}_j^{(t)} - \log(\gamma_i)\|^2 \\ + \frac{\rho}{2} \sum_{\substack{j \in \Omega_i \\ j > i}} \|\bar{a}_j^{(t)} + \log(\gamma_i)\|^2 + \frac{s_i}{\gamma_i}, \end{aligned} \quad (16)$$

where $\bar{a}_j^{(t)} = C_{i,j}^{(t)} + \log(\gamma_j^{(t)}) + \frac{\lambda_{i,j}^{(t)}}{\rho}$ and $\bar{a}_j^{(t)} = C_{j,i}^{(t)} - \log(\gamma_j^{(t)}) + \frac{\lambda_{j,i}^{(t)}}{\rho}$. The solution can be obtained by setting the gradient of the objective function in (16) to zero

$$\gamma_i^{(t+1)} = \frac{s_i}{1 + \rho |\Omega_i| - \rho \left(\sum_{j \in \Omega_i, j < i} \bar{a}_j^{(t)} - \sum_{j \in \Omega_i, j > i} \bar{a}_j^{(t)} \right)}, \quad (17)$$

where $|\Omega_i|$ denotes the cardinality of the set Ω_i .

2) \mathbf{C} -Update: The optimization problem with respect to \mathbf{C} can be decomposed into $N(N-1)/2$ convex subproblems as follows

$$C_{i,j}^{(t+1)} = \arg \min_{C_{i,j}} \frac{\beta_{i,j}}{M} |C_{i,j}| + \frac{\rho}{2} \left\| C_{i,j} - \bar{\gamma}_{i,j}^{(t+1)} + \frac{\lambda_{i,j}^{(t)}}{\rho} \right\|, \quad (18)$$

where $\bar{\gamma}_{i,j}^{(t+1)} = (\log(\gamma_i^{(t+1)}) - \log(\gamma_j^{(t+1)}))$. The expression in (18) admits a closed-form solution via a proximal gradient method [22] as

$$C_{i,j}^{(t+1)} = \eta_{\frac{\beta_{i,j}}{M\rho}} \left(\bar{\gamma}_{i,j}^{(t+1)} - \frac{\lambda_{i,j}^{(t)}}{\rho} \right), \quad (19)$$

where $\eta_{\kappa}(a) = \text{sign}(a) \max(|a| - \kappa, 0)$ is a soft-thresholding operator for any $a, \kappa \in \mathbb{R}_+$.

3) λ -Update: Finally, the dual variable is updated as

$$\lambda_{i,j}^{(t+1)} = \lambda_{i,j}^{(t)} + \rho \left(C_{i,j}^{(t+1)} - \bar{\gamma}_{i,j}^{(t+1)} \right), \forall i, j \quad (20)$$

The proposed solution is Algorithm 1. Despite having two loops (EM and ADMM), its overall complexity is minimally impacted since it requires empirically fewer EM iterations to converge compared to M-SBL.

¹We omit the EM iteration index in this subsection for sake of simplicity.

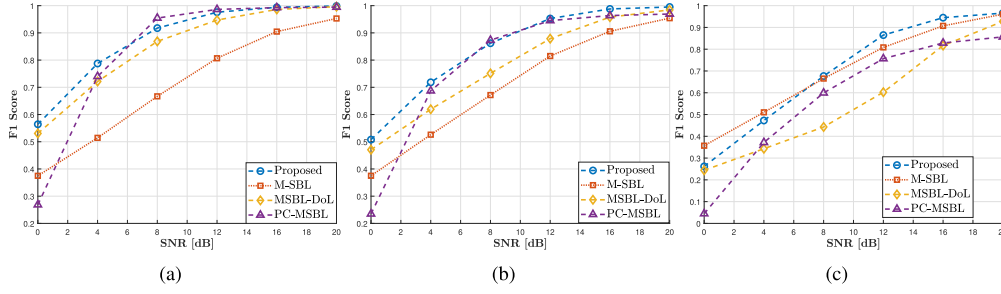


Fig. 1. Comparison of F1-score under varying signal-to-noise-ratio (SNR) levels and sparsity patterns: (a) block sparsity, (b) hybrid sparsity, and (c) isolated non-zero elements.

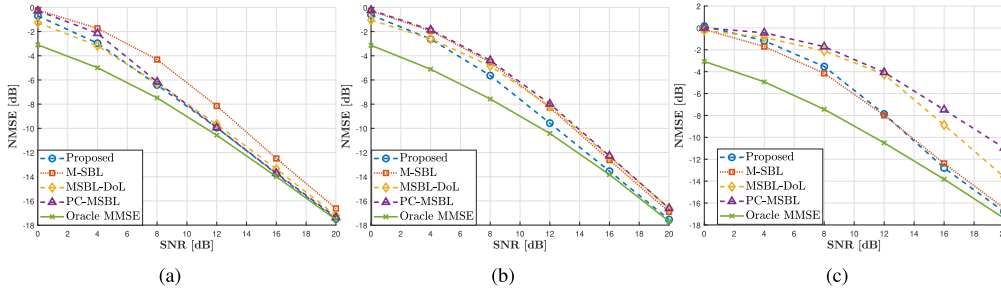


Fig. 2. Comparison of the signal reconstruction quality in terms of NMSE under varying SNR levels and sparsity patterns: (a) block sparsity, (b) hybrid sparsity, and (c) isolated non-zero elements.

IV. SIMULATION RESULTS

We consider a system setup with $N = 300$, $M = 5$, $L = 30$ and \mathbf{A} drawn from an i.i.d. complex Bernoulli distribution with normalized columns. The performance is evaluated in terms of the normalized mean square error (NMSE) given as $\frac{\mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2]}{\mathbb{E}[\|\mathbf{X}\|_F^2]}$, and the support recovery rate using the F1-Score $F1 = \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ as defined in [15]. We examine three sparsity use cases: 1) *block sparsity*: 5 non-zero blocks, each containing 5 elements, 2) *hybrid sparsity*: 3 active blocks, each containing 5 elements in addition to 5 randomly selected elements, and 3) *isolated sparsity*: 25 non-zero elements randomly selected non-zero elements. We compare the proposed algorithm with three baseline sparse recovery algorithms: 1) M-SBL [23], 2) MSBL-DoL [15], 3) PC-SBL [10]. In addition, we include the oracle minimum mean square error (MMSE) estimator to provide a baseline for NMSE performance.²

Figs. 1(a) and 2(a) depict the performances under the **block sparsity** pattern. The results show that algorithms explicitly designed to exploit block-sparse structure significantly outperform the standard M-SBL. These gains are attributed to the ability of these specialized algorithms to effectively exploit the inherent signal's sparse block structure. Note that PC-MSBL would outperform the proposed solution in some scenarios, which can be attribute to its hard-coupling mechanism that works well for block sparse scenarios.

The results for the **hybrid sparsity**, where multiple sparsity patterns coexist, can be found in Figs. 1(b) and 2(b). The proposed algorithm seen to significantly outperform the M-SBL, demonstrating its adaptability to complex signal structures. In

contrast, MSBL-DoL and PC-MSBL perform similarly to M-SBL, suggesting that their rigid coupling mechanisms limit their effectiveness in handling diverse sparsity patterns.

Finally, to demonstrate the flexibility and robustness of the proposed algorithm, its performance is also evaluated under **isolated sparsity** patterns. Figs. 1(c) and Fig. 2(c) show that the proposed algorithm achieves performance comparable to M-SBL, even in the presence of severe sparsity model mismatches. On the other hand, both PC-MSBL and MSBL-DoL exhibit significant degradation. The robustness of the proposed algorithm emerges from two key aspects of our adaptive prior $p(\gamma)$. Firstly, it learns the correlation structure directly from the data rather than imposing fixed coupling constraints. Secondly, it automatically adjusts the strength of the sparsity-promoting weights based on the observed signal statistics. This adaptive mechanism allows the algorithm to seamlessly handle both block-structured and isolated non-zero elements without requiring parameter tuning or prior knowledge of the sparsity pattern.

V. CONCLUSION

We proposed a novel SBL variant for block-sparse recovery in the MMV setup. The key innovations include an extended spatial coupling penalty that exploits block structure and a tuning-free learnable regularization weights to enhance robustness. We derived an efficient sequential closed-form ADMM-based solution for the evolved M-step. Numerical experiments demonstrate that our algorithm performs on par with state-of-the-art methods under standard sparsity assumptions and outperforms them when signals exhibit a block-sparse structure. This validates the algorithm's ability to effectively generalize to diverse signal structures. Our novel SBL variant represents an advancement in block-sparse signal recovery, with promising applications in areas such as mm-wave and near-field channel estimation.

²The simulation code and practical implementation details for our proposed algorithm are available on the following GitHub repository <https://github.com/Hamzadh/Adaptive-TV-M-SBL>

REFERENCES

- [1] M. Cui and L. Dai, "Channel estimation for extremely large-scale MIMO: Far-field or near-field?," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2663–2677, Apr. 2022.
- [2] X. Li, J. Fang, H. Li, and P. Wang, "Millimeter wave channel estimation via exploiting joint sparse and low-rank structures," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1123–1133, Feb. 2018.
- [3] H. Djelouat, M. J. Sillanpää, M. Leinonen, and M. Juntti, "Hierarchical MTC user activity detection and channel estimation with unknown spatial covariance," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 14620–14636, Oct. 2024.
- [4] E. Elhamifar and R. Vidal, "Block-sparse recovery via convex optimization," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4094–4107, Aug. 2012.
- [5] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [7] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.
- [8] 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, Jun. 2010.
- [9] Z. Zhang and B. D. Rao, "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, Apr. 2013.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] L. Wang, L. Zhao, L. Yu, J. Wang, and G. Bi, "Structured Bayesian learning for recovery of clustered sparse signal," *Signal Process.*, vol. 166, 2020, Art. no. 107255.
- [14] G. Nowak, T. Hastie, J. R. Pollack, and R. Tibshirani, "A fused lasso latent feature model for analyzing multi-sample aCGH data," *Biostatistics*, vol. 12, no. 4, pp. 776–791, 2011.
- [15] A. Sant, M. Leinonen, and B. D. Rao, "Block-sparse signal recovery via general total variation regularized sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1056–1071, 2022.
- [16] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [17] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver," *IEEE Trans. Inform. Theory*, vol. 67, no. 5, pp. 2925–2951, May 2021.
- [18] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.
- [19] 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.
- [20] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.
- [21] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5/6, pp. 877–905, 2008.
- [22] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," 2014, *arXiv:1411.3406*.
- [23] 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, Jul. 2007.