A Divide-and-conquer Approach for Sparse Recovery in High Dimensions

Bevelander, Aron; Batselier, Kim; Myers, Nitin Jonathan

DOI
10.1109/ICASSP49660.2025.10888687

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2025

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888687

A Divide-and-conquer Approach for Sparse Recovery in High Dimensions

Aron Bevelander, Kim Batselier, Nitin Jonathan Myers

Delft Center for Systems and Control, Delft University of Technology, The Netherlands

ABSTRACT

Block compressed sensing (BCS) alleviates the high storage and memory complexity with standard CS by dividing the sparse recovery problem into sub-problems. This paper presents a Welch bound-based guarantee on the reconstruction error with BCS, revealing that sparse recovery deteriorates with more partitions. To address this performance loss, we propose a data-driven BCS technique that leverages correlation across signal partitions. Our method surpasses classical BCS in moderate SNR regimes, with a modest increase in storage and computational complexities.

Index Terms— Block compressed sensing, computational complexity, memory limitation

1. INTRODUCTION

Compressed sensing (CS) recovers sparse high-dimensional signals from their compressed representation. The dimension of sparse signals in several applications is rapidly increasing with advances in sensing [1–3]. Real-time CS of high-dimensional signals requires intense memory and computational resources that may often be unavailable. Block compressed sensing (BCS), a class of CS, adopts a divide-and-conquer approach to reduce the storage and computational complexities needed for sparse recovery [4]. In BCS, the reconstruction problem is partitioned into sub-problems, where each sub-problem solves for a block within the sparse vector.

The sub-problems in BCS have a lower complexity than the original problem and they can be solved in parallel. Standard BCS methods typically obtain an equal number of CS measurements for each block, assuming a uniform distribution of sparse non-zero components across the blocks. In [5–8], however, varying numbers of measurements were acquired per block (also called as level) based on the sparsity in each block or the extent of coupling with other blocks. In [9], a permutation technique was proposed to "equally" distribute the non-zero components over different blocks. An important question in BCS is how partitioning the problem impacts reconstruction. An empirical study conducted in [10] showed that partitioning deteriorates reconstruction. To the best of our knowledge, reconstruction bounds with BCS as a function of the number of partitions or blocks have not been studied.

To address the performance loss with standard BCS, the sub-problems can be solved sequentially by using information from one reconstructed block as a prior for others, assuming correlation across blocks. Prior work on dynamical CS has exploited such correlation information in a different context than BCS. For instance, the techniques in [11, 12] use temporal snapshots as blocks within a high-dimensional signal. Additionally, [13] developed algorithms to leverage probabilistic information on the signal's support, which can be potentially derived from correlated blocks in BCS. While sequentially solving the sub-problems in BCS improves reconstruction, it sacrifices the ability to solve sub-problems in parallel.

In this paper, we study the limits of BCS by deriving a Welch bound-based recovery guarantee as a function of the number of partitions. Then, we develop a data-driven approach to learn the correlation across blocks. The learned correlation is then used for sparse recovery in our serial BCS method. Finally, we show that our data-driven serial BCS technique outperforms standard BCS at a moderate SNR.

2. BLOCK COMPRESSED SENSING

2.1. Preliminaries

In CS, a sparse signal $\mathbf{x} \in \mathbb{R}^n$ is projected on an m-dimensional space using a CS matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \ll n$. We define the compressed measurement of $\mathbf{x}$ as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v}, \tag{1}$$

where $\mathbf{v} \in \mathbb{R}^m$ is additive noise. CS algorithms estimate a sparse vector that best explains the measurements in $\mathbf{y}$, for a known $\mathbf{A}$. The computational complexity of CS algorithms in [14–16] is $\mathcal{O}(mn)$ for $o(1)$ sparse signals, as it is determined by matrix-vector multiplications with $\mathbf{A}$ and $\mathbf{A}^T$.

BCS uses a block-diagonal structure for $\mathbf{A}$ to alleviate the high memory and high computation associated with a generic CS matrix. We define β as the number of blocks or partitions in BCS, and assume that the blocks are of equal size. To allow discontiguous blocks in BCS, we define a permutation matrix $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$. A general form of the CS matrix in BCS is then

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_\beta \end{bmatrix} \mathbf{\Pi}, \tag{2}$$

where $\mathbf{A}_b \in \mathbb{R}^{\frac{m}{\beta} \times \frac{n}{\beta}}$ for $b \in \{1, 2, \cdots \beta\}$. Setting $\mathbf{\Pi}$ to an identity matrix $\mathbf{I}$ corresponds to standard BCS where the signal is partitioned into contiguous blocks. When the BCS matrix in (2) is used in (1), we observe that any measurement is a projection of a block within the sparse signal. We define $\mathbf{z} = \mathbf{\Pi x}$. The b^{th} block of $\mathbf{\Pi x}$, defined as $\mathbf{z}_b$, is a vector that contains entries of $\mathbf{\Pi x}$ indexed from $(b-1)n/\beta + 1$ to bn/β. The dimension of each signal block is n/β, and m/β compressed measurements are acquired per block using the CS matrix in (2).

We define $\mathbf{y}_b \in \mathbb{R}^{\frac{m}{\beta}}$ as the measurements of the b^{th} block and $\mathbf{v}_b$ as the associated noise. Then, $\mathbf{y} = (\mathbf{y}_1; \mathbf{y}_2; \cdots; \mathbf{y}_\beta)$ is a column stacked version of the block measurements and $\mathbf{v} = (\mathbf{v}_1; \mathbf{v}_2; \cdots; \mathbf{v}_\beta)$. When the BCS matrix in (2) is used in (1), the block measurements can be written as

$$\mathbf{y}_b = \mathbf{A}_b \mathbf{z}_b + \mathbf{v}_b \ \ \forall b \in \{1, 2, \cdots, \beta\}. \tag{3}$$

To estimate $\mathbf{x}$, a straightforward approach is to first solve for $\{\mathbf{z}_b\}_{b=1}^{\beta}$ from (3). These problems can be solved in parallel. Then, the reconstructed blocks can be stacked together to obtain $\hat{\mathbf{z}}$ and $\mathbf{x}$ can be estimated as $\hat{\mathbf{x}} = \mathbf{\Pi}^{-1}\hat{\mathbf{z}}$.

Due to the block structure in BCS matrices, it is sufficient to store $\mathcal{O}(\beta \cdot mn/\beta^2)$ entries of (2). This requirement is β times lower compared to standard CS, which necessitates storing $\mathcal{O}(mn)$ entries. Furthermore, the complexity of matrix-vector multiplications in solving for one block in (3) is $\mathcal{O}(mn/\beta^2)$, when compared to $\mathcal{O}(mn)$ for standard CS. When the β blocks are independently recovered, BCS has a computational complexity $1/\beta^2$ lower than standard CS. If blocks are estimated sequentially, BCS achieves a computational complexity $1/\beta$ lower than standard CS.

2.2. Mutual coherence-based performance limits of BCS

The mutual coherence of $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as [17]

$$\mu(\mathbf{A}) = \max_{(k,\ell):k \neq \ell} \frac{|\langle \mathbf{a}_k, \mathbf{a}_\ell \rangle|}{\|\mathbf{a}_k\|_2 \|\mathbf{a}_\ell\|_2}, \tag{4}$$

where $\mathbf{a}_k$ is the k^{th} column of $\mathbf{A}$. A small mutual coherence allows for better sparse reconstruction with the orthogonal matching pursuit (OMP), a greedy CS algorithm [17]. For a CS matrix of size $m \times n$, the smallest mutual coherence that can be achieved is given by the Welch bound [18], i.e.,

$$\mu(\mathbf{A}) \geq \sqrt{\frac{n-m}{m(n-1)}}. \tag{5}$$

The bound in (5) only depends on the size of the CS matrix.

We derive a lower bound on the mutual coherence of the BCS matrix in (2). We first observe that permuting the columns of $\mathbf{A}$ does not affect mutual coherence, so we set $\mathbf{\Pi}$ to an identity matrix for our analysis. Next, we note that columns of $\mathbf{A}$ from different blocks are orthogonal. For example, the first column of $\mathbf{A}$ in (2) is orthogonal to columns indexed from $n/\beta + 1$ to n. Thus, for a BCS matrix, the mutual coherence in (4) is not determined by a pair of columns chosen from two different blocks. Applying the definition in (4) for the BCS matrix in (2) results in

$$\mu(\mathbf{A}) = \max\{\mu(\mathbf{A}_1), \mu(\mathbf{A}_2), \cdots, \mu(\mathbf{A}_\beta)\}. \tag{6}$$

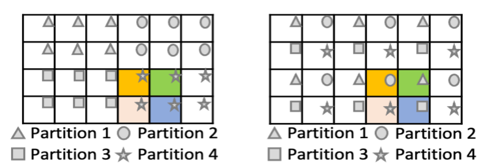
As each of the β mutual coherences in (6) is lower bounded by the Welch bound associated with an $m/\beta \times n/\beta$ matrix,

$$\mu(\mathbf{A}) \geq \sqrt{\frac{\frac{n}{\beta} - \frac{m}{\beta}}{\frac{m}{\beta}(\frac{n}{\beta} - 1)}} = \sqrt{\frac{n-m}{m(\frac{n}{\beta} - 1)}}. \tag{7}$$

For $\beta > 1$, we observe that our lower bound in (7) for BCS is larger than the Welch bound in (5). This is because BCS matrices cannot achieve the Welch bound in (5) due to the block diagonal constraint on the CS matrix. Our coherence limit in (7) can be used to study the MSE bound with the OMP. For an s-sparse signal, the mean squared error (MSE) with the OMP can be bounded as $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 \leq 2(1+\alpha)s\sigma^2 \log m/((1-(s-1)\mu)^2)$, for a constant α [17]. By substituting our coherence limit from (7) in this bound, we observe from Fig. 3a that the MSE bound with the OMP increases with β, indicating that partitioning the CS problem leads to poor reconstruction.

3. DATA-DRIVEN SERIAL BCS

To alleviate the poor MSE issue with partitioning in BCS, we exploit correlation across the blocks within the sparse signal during reconstruction. We illustrate our approach to exploit these correlations using a 2D clustered sparse signal shown in Fig. 1. Our data-driven serial BCS method, however, is discussed for the more general case of tensors that exhibit clustered sparsity.



(a) Contiguous partitions (b) Comb-like partitions

Fig. 1: Examples of partitioning strategies ($\beta = 4$) for a 2D sparse signal. The non-zero entries of the 2D sparse signal here are shaded. We observe from this example that when clustered sparse signals are partitioned with a comb-like pattern, reconstructing one block provides useful side information on the support of other blocks.

A good partitioning strategy is key to the success of standard BCS and serial BCS. Contiguous partitioning (shown in Fig. 1a), which groups contiguous indices into a block, is ineffective for clustered sparse signals. This approach often concentrates non-zero coefficients in a few blocks, leading to poor reconstruction when an equal number of CS measurements are acquired per block. Comb-like periodic partitioning (shown in Fig. 1b) is better suited for clustered sparse signals as it distributes non-zero coefficients more evenly across

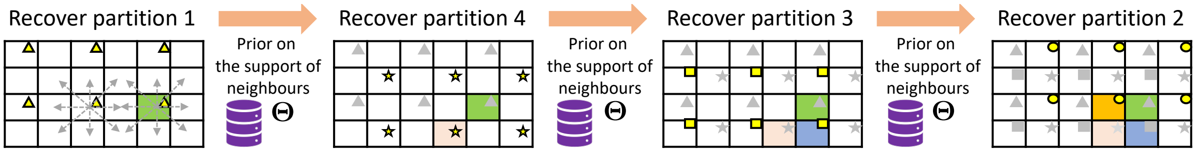


Fig. 2: Our proposed data-driven serial BCS method first reconstructs one of the signal blocks (partitions). Then, the reconstructed signal together with the learned support correlation $\mathbf{\Theta}$ is used to find a prior on the support associated with the other blocks. This prior is used together with the BCS measurements to reconstruct the next block, and the process is repeated until all the blocks are recovered.

blocks. We observe from Fig. 1b that a non-zero entry reconstructed in a comb-like block (e.g., entry at Δ) provides useful side-information on the support of its neighbours, under the clustered sparsity assumption. This information can be used as a prior to recover other blocks.

We discuss an offline algorithm to learn side-information provided by a non-zero component on its neighbours' support. Focusing on clustered sparse signals and comb-like partitions, our algorithm uses a dataset of sparse signals $\{\boldsymbol{\mathcal{X}}^{(j)}\}_{j=1}^{J}$ to estimate the probability that a neighbour is non-zero, given an entry is non-zero. This probability is calculated by counting the instances of non-zero neighbours in the dataset. For a 2D signal, a naive approach finds $\sim 8n$ such probabilities as there are 8 neighbours around each entry except for those at the edges. For a d^{th} order tensor, this number increases to $\sim (3^d-1)n$. As storing all these values consumes significant memory, we assume spatially invariant probabilities and learn a support correlation kernel with 3^d-1 entries for a d^{th} order tensor. Our procedure to learn this kernel $\mathbf{\Theta}$ from a dataset of sparse signals is summarized in Algorithm 1.

Algorithm 1 Construction of the support correlation kernel

Input: Sparse signals $\{\boldsymbol{\mathcal{X}}^{(j)}\}_{j=1}^{J}$ in $\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$.
Define: Set $\mathcal{S}_j$ has indices of non-zero entries in $\boldsymbol{\mathcal{X}}^{(j)}$.
  for $j: 1\rightarrow J$ **do**
    Initialize d^{th} order kernel $\boldsymbol{\kappa}=\mathbf{0}\in\mathbb{R}^{3\times 3\times\cdots(d\,\text{times})}$.
    for index $(\omega_1,\omega_2,\cdots,\omega_d)$ in $\mathcal{S}_j$ **do**
      Find immediate neighbours around this index:
      $\Omega=\{(\zeta_1,\zeta_2,\cdots,\zeta_d): |\omega_i-\zeta_i|\leq 1\forall i\in[d]\}$.
      Scan through all the entries of $\boldsymbol{\mathcal{X}}^{(j)}$ at Ω
        If $\boldsymbol{\mathcal{X}}^{(j)}(\zeta_1,\zeta_2,\cdots,\zeta_d)\neq 0$:
        Add 1 to $\boldsymbol{\kappa}(\zeta_1-\omega_1,\zeta_2-\omega_2,\cdots,\zeta_d-\omega_d)$.
      Set $\boldsymbol{\kappa}(0,0,\cdots,0)=0$.
    end for
    $\mathbf{\Theta}^{(j)}=\boldsymbol{\kappa}/\text{cardinality}(\mathcal{S}_j)$
  end for
  return $\mathbf{\Theta}=\sum_{j=1}^{J}\mathbf{\Theta}^{(j)}/J$

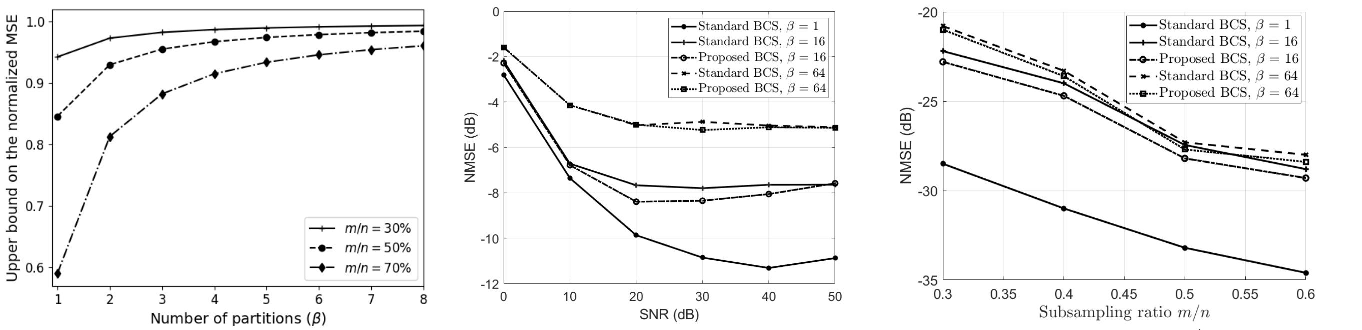
We discuss notation involved in our data-driven BCS method to recover a sparse tensor $\boldsymbol{\mathcal{X}}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ from its compressed version. Here, the dimension of the signal is $n=n_1\cdot n_2\cdots n_d$. Assuming that the ℓ^{th} mode is partitioned by a factor of β_ℓ, the total number of partitions is $\beta=\beta_1\cdot\beta_2\cdots\beta_d$. The tensor $\boldsymbol{\mathcal{X}}_b\in\mathbb{R}^{\frac{n_1}{\beta_1}\times\frac{n_2}{\beta_2}\times\cdots\times\frac{n_d}{\beta_d}}$ comprises the entries of $\boldsymbol{\mathcal{X}}$ at the indices in partition b. Further, we define $\boldsymbol{\mathcal{X}}_{b,\text{ext}}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ as a tensor that has the entries of $\boldsymbol{\mathcal{X}}$ at the indices in partition b and zeros at all the other indices. We define $\boldsymbol{\mathcal{P}}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ such that $\boldsymbol{\mathcal{P}}(\zeta_1,\zeta_2,\cdots,\zeta_d)$ is proportional to the probability that $\boldsymbol{\mathcal{X}}(\zeta_1,\zeta_2,\cdots,\zeta_d)$ is non-zero. After each stage of our algorithm that solves for a block within $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{P}}$ is updated using the reconstructed block and the support correlation kernel $\mathbf{\Theta}$.

Our data-driven BCS technique, summarized in Algorithm 2, first solves for one of the β blocks within the sparse signal from its compressed representation. Without loss of generality, our algorithm first reconstructs $\boldsymbol{\mathcal{X}}_1$ from $\mathbf{y}_1=\mathsf{A}_1(\boldsymbol{\mathcal{X}}_1)+\mathbf{v}_1$, where $\mathsf{A}_1(\cdot)$ denotes a linear compression operator. To solve for this first block, we set $\boldsymbol{\mathcal{P}}=s\mathbf{1}/n$, where s is the average sparsity level estimated from the dataset. We use the logit-weighted OMP (LW-OMP) algorithm [13], that exploits the support prior $\boldsymbol{\mathcal{P}}$ together with the compressed measurements, to reconstruct the signal. The reconstructed block $\hat{\boldsymbol{\mathcal{X}}}_1$ is extended with zeros to obtain $\hat{\boldsymbol{\mathcal{X}}}_{1,\text{ext}}$, which is then convolved with $\mathbf{\Theta}$ to update the support prior. The updated prior $\boldsymbol{\mathcal{P}}$ is then used to determine the block for which the most side-information on its support is available, i.e., the block for which $\text{sum}\{\text{vec}(\boldsymbol{\mathcal{P}}_b)\}$ is maximum. This block is recovered next and the procedure is repeated until all the blocks within $\boldsymbol{\mathcal{X}}$ are recovered. Our algorithm for the matrix case ($d=2$) is illustrated in Fig. 2.

Algorithm 2 Proposed data-driven serial BCS algorithm

Input: BCS measurements $\{\mathbf{y}_b\}_{b=1}^{\beta}$, CS operators $\{\mathsf{A}_b\}_{b=1}^{\beta}$, Learned support correlation kernel $\mathbf{\Theta}$.
Define: Set $\nu=[\beta]$, Support prior tensor $\boldsymbol{\mathcal{P}}=s\mathbf{1}/n$ (s is the average sparsity level), $\hat{\boldsymbol{\mathcal{X}}}=\mathbf{0}$ and $b=1$.
  for $j: 1\rightarrow\beta$ **do**
    $\hat{\boldsymbol{\mathcal{X}}}_b=\text{LW-OMP}(\mathbf{y}_b,\mathsf{A}_b,\boldsymbol{\mathcal{P}}_b,\sigma^2)$ #Solve for block b
    $\hat{\boldsymbol{\mathcal{X}}}=\hat{\boldsymbol{\mathcal{X}}}+\hat{\boldsymbol{\mathcal{X}}}_{b,\text{ext}}$
    $\boldsymbol{\mathcal{P}}=\boldsymbol{\mathcal{P}}+|\hat{\boldsymbol{\mathcal{X}}}_{b,\text{ext}}\circledast\mathbf{\Theta}|$ #Update prior using estimate
    $\nu\leftarrow\nu\setminus\{b\}$
    $b=\arg\max_{k\in\nu}[\text{sum}\{\text{vec}(\boldsymbol{\mathcal{P}}_k)\}]$ #Next block to solve
  end for
  return $\hat{\boldsymbol{\mathcal{X}}}$

Our data-driven serial BCS with LW-OMP exploits structure across partitions but incurs higher computational complexity than standard BCS with OMP. This is because our method serially solves for the β partitions, unlike standard



(a) NMSE bound using (7) (b) NMSE with SNR (c) NMSE with m/n

Fig. 3: For $s = 50$, $n = 10^4$, $\alpha = 0.5$, and $\sigma = 10^{-2}$, we see from Fig. 3a that the OMP bound using (7) increases with β. We use the near-field dataset for Fig. 3b and Fig. 3c. For a subsampling ratio of 40%, Fig. 3b shows that the proposed method outperforms standard BCS in the moderate SNR regime. For SNR $= 30\,\text{dB}$, we see from Fig. 3c that NMSE decreases with the subsampling ratio m/n for any β.

BCS which can be parallelized. Further, support prior update in our method requires convolution, which adds a complexity of $\mathcal{O}(3^d n/\beta)$ per partition. Considering β partitions, the increase in complexity with our method over serial BCS is $\mathcal{O}(3^d n)$, still modest relative to unpartitioned standard CS.

4. SIMULATION RESULTS

We consider sparse 4D spatial near-field channel estimation with a 16×16 transmitter (TX) and a 8×8 receiver (RX). The TX and the RX use half-wavelength spaced uniform planar arrays at $300\,\text{GHz}$. For a transceiver distance of $30\,\text{cm}$, we generate several channel realizations according to [19], by translating the RX and also rotating it at random. Each channel realization is a tensor in $\mathbb{C}^{16\times16\times8\times8}$. We use $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{16\times16\times8\times8}$ to denote the 4D-discrete Fourier transform of the channel, which exhibits clustered sparsity. The kernel $\boldsymbol{\Theta}$ was learned with Algorithm 1 using these realizations.

The measurements in BCS were acquired by applying the comb-like codes in [20] at both the TX and the RX. By adjusting the periodicity within the comb, the number of partitions β was configured. In our simulations, we consider $\beta \in \{1, 16, 64\}$ to compare the performance of the proposed approach against standard BCS. Note that $\beta = 1$ corresponds to the unpartitioned problem, which is solved using the standard OMP algorithm. Our approach employs the LW-OMP [13] within Algorithm 2, whereas standard BCS uses the classical OMP algorithm to solve for each partition.

To evaluate the performance of BCS algorithms, we use the normalized mean squared error (NMSE) between the sparse channel and its estimate, i.e., $\mathbb{E}[\|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_{\mathrm{F}}^2]/\mathbb{E}[\|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2]$, where the expectation is taken across all channel realizations. The dimension of the sparse vector solved for $\beta = 1$ is $16 \times 16 \times 8 \times 8 = 16384$. With BCS, however, this dimension reduces to $16384/\beta$ per partition. From Fig. 3b, we observe that BCS results in a poor NMSE than standard CS (equivalent to BCS with $\beta = 1$). Moreover, the NMSE deteriorates for increasing β, aligning with our analysis in Sec. 2.2, where a higher β led to an increased error bound.

Number of blocks β	Standard BCS	Proposed method
1	18.2×10^8ms	N/A for $\beta = 1$
16	5910 ms	9827 ms
64	476 ms	1469 ms

Table 1: Computation time with standard BCS and the proposed method. BCS with $\beta = 1$ is same as classical CS.

We notice from Fig. 3b that the proposed data-driven serial BCS outperforms standard BCS in the moderate SNR regime. At low SNR, the performance of both the methods is almost the same. This is because of the noise in the reconstruction, which impacts the support prior estimated, i.e., $\boldsymbol{\mathcal{P}}$, in Algorithm 2. As the support prior estimate is not reliable at low SNR, it does not improve the reconstruction with LW-OMP. At high SNR, the measurements in each block provide reliable information to determine the support even without any prior information. Next, we observe from Fig. 3c that the proposed method outperforms standard BCS for different subsampling ratios, i.e., m/n, at an SNR of $30\,\text{dB}$. This performance improvement comes at the expense of an increased computational complexity compared to standard BCS, owing to the support prior calculation in our approach. The increase in complexity, however, is small when compared to the complexity of the unpartitioned CS problem for $\beta = 1$. The computation times of all the methods for $n = 16384$ and a subsampling ratio of $\sim 40\%$ is summarized in Table 1.

5. CONCLUSIONS

In this paper, we studied block compressed sensing (BCS) for high-dimensional sparse recovery. We proved that the lower bound on the mutual coherence of a BCS matrix is higher than the Welch bound associated with a standard CS matrix of the same dimensions. We also proposed a data-driven serial BCS method, which uses reconstructed blocks to estimate priors on the support of subsequent blocks. Using simulations for a sparse near-field channel estimation problem, we showed that our method outperforms standard BCS at moderate SNR.

6. REFERENCES

[1] G. Calisesi, A. Ghezzi, D. Ancora, C. D'Andrea, G. Valentini, A. Farina, and A. Bassi, "Compressed sensing in fluorescence microscopy," *Progress in Biophysics and Molec. Biol.*, vol. 168, pp. 66–80, 2022.

[2] W. L. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, and D. M. Mittleman, "A single-pixel terahertz imaging system based on compressed sensing," *Applied Physics Letters*, vol. 93, no. 12, 2008.

[3] S. Nie and I. F. Akyildiz, "Deep kernel learning-based channel estimation in ultra-massive MIMO communications at 0.06-10 THz," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[4] L. Gan, "Block compressed sensing of natural images," in *2007 15th Intl. Conf. on Digital Signal Process.*, 2007, pp. 403–406.

[5] J. Chen, X. Zhang, and H. Meng, "Self-adaptive sampling rate assignment and image reconstruction via combination of structured sparsity and non-local total variation priors," *Digital Signal Process.*, vol. 29, pp. 54–66, 2014.

[6] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 08, p. P08009, 2012.

[7] B. Adcock, A. Hansen, B. Roman, and G. Teschke, "Generalized sampling: stable reconstructions, inverse problems and compressed sensing over the continuum," in *Advances in imaging and electron physics*. Elsevier, 2014, vol. 182, pp. 187–279.

[8] A. Bastounis and A. C. Hansen, "On random and deterministic compressed sensing and the restricted isometry property in levels," in *Proc. of the IEEE Intl. Conf. on Sampling Theory and Appl. (SampTA)*, 2015, pp. 297–301.

[9] B. Zhang, Y. Liu, J. Zhuang, K. Wang, and Y. Cao, "Matrix permutation meets block compressed sensing," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 69–78, 2019.

[10] R. Pournaghshband and M. Modarres-Hashemi, "A novel block compressive sensing algorithm for sar image formation," *Signal Processing*, vol. 210, p. 109053, 2023.

[11] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, 2013.

[12] M. S. Asif and J. Romberg, "Dynamic updating for sparse time varying signals," in *Annual Conf. on Inform. Sciences and Sys.*, 2009, pp. 3–8.

[13] J. Scarlett, J. Evans, and S. Dey, "Compressed sensing with prior information: Information-theoretic limits and practical decoders," *IEEE Transactions on Signal Processing*, vol. 61, p. 427, 01 2013.

[14] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Inform. theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[15] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. and comput. harmonic anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[16] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[17] Z. Ben-Haim, Y. C. Eldar, and M. Elad, "Coherence-based performance guarantees for estimating a sparse vector under random noise," *IEEE Trans. on Signal Process.*, vol. 58, no. 10, pp. 5030–5043, 2010.

[18] L. Welch, "Lower bounds on the maximum cross correlation of signals," *IEEE Trans. on Inform. theory*, vol. 20, no. 3, pp. 397–399, 1974.

[19] E. Torkildson, U. Madhow, and M. Rodwell, "Indoor millimeter wave MIMO: Feasibility and performance," *IEEE Trans. on Wireless Commun.*, vol. 10, pp. 4150–4160, 12 2011.

[20] H. Masoumi, M. Verhaegen, and N. J. Myers, "In-sector compressive beam acquisition for MmWave and THz radios," *IEEE Trans. on Commun.*, 2024.