

A Hierarchical View of Structured Sparsity in Kronecker Compressive Sensing

Yanbin He and Geethu Joseph

Signal Processing Systems Group, Delft University of Technology, The Netherlands

Emails: {y.he-1,g.joseph}@tudelft.nl

Abstract—Kronecker compressed sensing refers to using Kronecker product matrices as sparsifying bases and measurement matrices in compressed sensing. This work focuses on the Kronecker compressed sensing problem, encompassing three sparsity structures: (i) a standard sparsity model with arbitrarily positioned nonzero entries, (ii) a hierarchical sparsity model where nonzero entries are concentrated in a few blocks, each with only a subset of nonzero entries, and (iii) a Kronecker-supported sparsity model where the support vector is a Kronecker product of smaller vectors. We present a hierarchical view of Kronecker compressed sensing that explicitly reveals a multiple-level sparsity pattern. This framework allows us to utilize the Kronecker structure of dictionaries and design a two-stage sparse recovery algorithm for different sparsity models. Further, we analyze the restricted isometry property of Kronecker-structured matrices under different sparsity models. Simulations show that our algorithm offers comparable recovery performance to state-of-the-art methods while significantly reducing runtime.

Index Terms—Compressed sensing, sparse Bayesian learning, restricted isometry property, hierarchical sparsity

I. INTRODUCTION

Multidimensional signals arise in various applications such as image processing [1]–[4] and wireless communications [5]–[7], representing data across multiple dimensions. For example, images are intrinsically two-dimensional arrays (matrices) [8], and wireless signals are dependent on the signal’s angles of departure and arrival [7], [9]. Moreover, these signals admit sparse representations in an appropriate basis, such as the discrete Fourier basis for images and angular domain in communications, enabling compressed sensing-based acquisition and recovery. Due to the physical nature of multidimensional signals, it is natural to measure the signal dimension-wise. This induces the Kronecker compressed sensing, where a Kronecker product matrix is utilized to characterize different dimensions of multidimensional signal [1]. A fundamental signal model in the Kronecker compressed sensing is

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} = (\mathbf{H}_1 \otimes \mathbf{H}_2)\mathbf{x} + \mathbf{n}, \quad (1)$$

where $\mathbf{H}_1 \in \mathbb{R}^{M_1 \times N_1}$, $\mathbf{H}_2 \in \mathbb{R}^{M_2 \times N_2}$, $\mathbf{H} \in \mathbb{R}^{\bar{M} \times \bar{N}}$, $\mathbf{x} \in \mathbb{R}^{\bar{N}}$, $\mathbf{y} \in \mathbb{R}^{\bar{M}}$, and \mathbf{n} is the noise, with $\bar{M} = M_1 M_2$ and $\bar{N} = N_1 N_2$. The goal is to recover the sparse vector \mathbf{x} from noisy measurement \mathbf{y} given the Kronecker product dictionary $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$.

Furthermore, in many applications, sparse vector \mathbf{x} can also exhibit additional structures. One such structure is *hierarchically structured* sparsity wherein the sparse vector \mathbf{x} in (1) is partitioned into N_1 blocks, each of length N_2 . The vector \mathbf{x} follows the (s_1, s_2) -hierarchical sparsity if only s_1 among

the N_1 blocks are nonzero and each of these non-zero blocks is s_2 -sparse. This structure commonly appears in channel estimation for massive multiple-input multiple-output systems [10]. Further, if all nonzero blocks share a common support, the sparse vector is said to exhibit the (s_1, s_2) -Kronecker-supported sparsity. This term comes from the fact that the support of \mathbf{x} can be expressed as the Kronecker product of two binary support vectors. This structure appears in wireless communications [9], [11], [12] and image processing [2]–[4]. Motivated by the different sparsity patterns in Kronecker compressed sensing, we focus on recovering \mathbf{x} from \mathbf{y} in (1), leveraging the prior knowledge of its sparsity pattern and the Kronecker matrix \mathbf{H} . Particularly, we examine three models: the standard sparsity, hierarchical sparsity, and Kronecker-supported sparsity models.

Research on sparse recovery with Kronecker compressed sensing often focuses on algorithms and guarantees. Standard sparsity models typically employ ℓ_1 -norm-based algorithms [13] or greedy algorithms [14], that do not exploit the Kronecker structure of \mathbf{H} . However, some specialized algorithms have been designed for hierarchical sparsity and Kronecker-supported sparsity models. A notable example is a hard-thresholding pursuit-based algorithm for hierarchical sparsity model [10]. While it explicitly enforces (s_1, s_2) -hierarchical sparsity using a specific thresholding operator, it does not exploit the Kronecker structure of \mathbf{H} and requires the true value of s_1 and s_2 . For Kronecker-supported sparsity models, there are multiple approaches such as greedy algorithm [3] and Bayesian algorithms [7], [15]. The state-of-the-art is a sparse Bayesian learning (SBL)-based algorithm for Kronecker-supported sparse recovery [7]. This work explicitly enforces the Kronecker-structured support of \mathbf{x} and leverages the Kronecker structure of \mathbf{H} to reduce complexity [9]. However, the algorithm still suffers from high complexity. Thus, we aim to design a generalized framework with more efficient algorithms by fully exploiting the Kronecker structure of \mathbf{H} .

Apart from algorithm development, theoretical guarantees have also been studied for the Kronecker compressed sensing, which mostly focuses on the *restricted isometry property* (RIP)-based analysis [16]. The standard RIP analysis establishes the robust recovery of standard compressed sensing algorithms like basis pursuit and iterative hard thresholding. The RIP analysis is also extended to Kronecker-structured dictionaries for standard sparsity [1], [13], [17] and hierarchical sparsity [10]. However, the RIP analysis of Kronecker-

supported sparse vectors is missing in the literature. Further, there is no unified discussion for the RIP of Kronecker-structured matrices regarding sparse vectors with patterns. Our framework provides an integrated approach for the RIP analysis for Kronecker product matrix \mathbf{H} with respect to sparse vector \mathbf{x} with different patterns.

In this paper, we present a *hierarchical view* on Kronecker compressed sensing, capturing its dimension-wise signal acquisition and explicitly revealing multi-level sparsity patterns. Our contributions are twofold.

- *Algorithm Design*: We design a two-stage sparse recovery algorithm that leverages the Kronecker structure of the dictionary via the hierarchical view, reducing complexity while maintaining competitive recovery performance.
- *Theoretical Analyses*: Using the hierarchical view, we re-examine the RIP analysis for Kronecker compressed sensing, providing a generalized result that encompasses structured sparsity, with the three sparsity models as special cases.

Overall, we introduce a new perspective that bridges different sparsity models in Kronecker compressed sensing, leading to efficient algorithms and unified analysis.

II. HIERARCHICAL VIEW AND TWO-STAGE SPARSE RECOVERY ALGORITHM

This section explores the hierarchical view of Kronecker compressed sensing and a recovery approach based on it.

A. Hierarchical View

Our hierarchical view relies on the structure of the measurement system in the Kronecker compressed sensing model (1), where the Kronecker-structured matrix \mathbf{H} has two factor matrices, \mathbf{H}_1 and \mathbf{H}_2 . These factor matrices operate at different levels: \mathbf{H}_1 captures block-level while \mathbf{H}_2 focuses on intra-block, following a hierarchical structure. To illustrate this view, we first partition \mathbf{x} into N_1 blocks of size N_2 . Let the i th block be denoted by $\mathbf{x}_i \in \mathbb{R}^{N_2}$. Then, we can rearrange \mathbf{x} into $\mathbf{X} \in \mathbb{R}^{N_2 \times N_1}$ such that the i th column of \mathbf{X} is \mathbf{x}_i . Similarly, we can rearrange \mathbf{y} and \mathbf{n} into matrices $\mathbf{Y} \in \mathbb{R}^{M_2 \times M_1}$ and $\mathbf{N} \in \mathbb{R}^{M_2 \times M_1}$, such that $\text{vec}(\mathbf{Y}) = \mathbf{y}$ and $\text{vec}(\mathbf{N}) = \mathbf{n}$, respectively. Here, $\text{vec}(\cdot)$ denotes vectorization. Since $\text{vec}(\mathbf{H}_2 \mathbf{X} \mathbf{H}_1^\top) = (\mathbf{H}_1 \otimes \mathbf{H}_2) \mathbf{x}$, we obtain

$$\mathbf{Y}^\top = \mathbf{H}_1 (\mathbf{H}_2 \mathbf{X})^\top + \mathbf{N}^\top, \quad (2)$$

where the i th row of \mathbf{Y} represents the i th row of $\mathbf{H}_2 \mathbf{X}$ measured by \mathbf{H}_1 . Also, the i th row of $\mathbf{H}_2 \mathbf{X}$ corresponds to the block \mathbf{x}_i measured by \mathbf{H}_2 . Therefore, \mathbf{H}_1 measures at a higher level by operating on the rows of $\mathbf{H}_2 \mathbf{X}$, effectively capturing the sparsity structure at the block level.

The above perspective can also be interpreted directly from (1). Recall that the Kronecker product matrix \mathbf{H} possesses a column-block structure with a repetitive block pattern along its columns. Here, each block of columns is obtained by taking the Kronecker product of a column of \mathbf{H}_1 with \mathbf{H}_2 . Also, the column-block structure of \mathbf{H} matches with the blocks of \mathbf{x} . Hence, in this hierarchical framework, \mathbf{H}_2 first measures each block of \mathbf{x} . The resulting measurements of all blocks

are then processed by \mathbf{H}_1 , which captures information at a higher, global level. The relation (2) explicitly captures this measurement model, where intra-block measurement by \mathbf{H}_2 is followed by block-level measurement by \mathbf{H}_1 .

B. Algorithm Development

The hierarchical view in the Kronecker compressed sensing problem (1) indicates that the sparse vector \mathbf{x} can also be recovered in a hierarchical manner, leading to a two-stage recovery approach, as discussed next.

The first step of the algorithm treats $(\mathbf{H}_2 \mathbf{X})^\top$ as unknown and estimates it by solving (2). Also, $\mathbf{H}_2 \mathbf{X}$ exhibits a column-wise sparsity pattern, i.e., a nonzero column of $\mathbf{H}_2 \mathbf{X}$ corresponds to a nonzero block of \mathbf{x} while a zero column corresponds to a zero block. Thus, $(\mathbf{H}_2 \mathbf{X})^\top$ is a row sparse matrix and recovering $(\mathbf{H}_2 \mathbf{X})^\top$ from (2) can be formulated as a multiple measurement vector (MMV) problem. It can be solved using any MMV variants of compressed sensing algorithms, such as orthogonal matching pursuit (OMP) or SBL.

Let the estimate of $(\mathbf{H}_2 \mathbf{X})^\top$ after the first step be $\tilde{\mathbf{X}}$. In the next step, we treat $\tilde{\mathbf{X}}$ as measurements and recover \mathbf{X} from

$$\tilde{\mathbf{X}}^\top = \mathbf{H}_2 \mathbf{X} + \tilde{\mathbf{N}}, \quad (3)$$

where $\tilde{\mathbf{N}}$ represents noise. For standard and hierarchical sparsity models, the support of the different blocks of \mathbf{x} (or columns of \mathbf{X}) are different. So, we treat problem (3) as multiple independent single measurement vector (SMV) problems, which can be solved either sequentially or in parallel using any standard compressed sensing algorithm. Nonetheless, for the Kronecker-supported sparsity model, problem (3) is an MMV problem because the support is common across different blocks. The resulting algorithm, named Two-Stage Recovery (TSR), is summarized in Algorithm 1.

Algorithm 1 Two-stage sparse recovery

Input: Measurement \mathbf{y} , dictionaries $\mathbf{H}_1 \in \mathbb{R}^{M_1 \times N_1}$, and $\mathbf{H}_2 \in \mathbb{R}^{M_2 \times N_2}$

- 1: Re-order \mathbf{y} to obtain \mathbf{Y}
- 2: Solve (2) to obtain $\tilde{\mathbf{X}}$ using any MMV algorithm
- 3: Solve (3) for \mathbf{X} using any recovery algorithm (use MMV variant for Kronecker-supported sparsity model)

Output: Sparse vector $\mathbf{x} = \text{vec}(\mathbf{X})$

We further analyze the complexity of our TSR algorithm to demonstrate the benefit of exploiting the Kronecker structure of \mathbf{H} via the hierarchical view. We consider TSR combined with SBL [18] and MMV-SBL [19], and HTP [20] as sparse recovery algorithms, referred to as TSSBL, TSMSBL, and TSHTP, respectively. Assume M_1 and M_2 are $\mathcal{O}(M)$, N_1 and N_2 are $\mathcal{O}(N)$, and $M < N$. Table I compares the time and space complexity of our algorithms with other state-of-the-art algorithms, including SBL for standard sparsity, HiHTP [10] for (s_1, s_2) -hierarchical sparsity, and AM- and SVD-KroSBL [7] for Kronecker-supported sparsity. The results against SBL-based methods indicate that TSSBL exhibits lower time and

TABLE I: Complexity of different algorithms in different sparse recovery problems.

Method	Time Complexity	Space Complexity
Recovery of s -sparse vectors		
TSSBL	$\mathcal{O}(N^3M)$	$\mathcal{O}(N^2)$
SBL	$\mathcal{O}(N^4M^2)$	$\mathcal{O}(N^4)$
Recovery of (s_1, s_2) -hierarchically sparse vectors		
TSSBL	$\mathcal{O}(N^3M)$	$\mathcal{O}(N^2)$
TSHTP (sequential)	$\mathcal{O}(MN^2 + s_1M^2 + s_2M^2N)$	$\mathcal{O}(MN)$
TSHTP (parallel)	$\mathcal{O}(M^2N + (s_1 + s_2)M^2)$	$\mathcal{O}(MN^2)$
HiHTP [10]	$\mathcal{O}(s_1s_2M^4 + (MN)^2)$	$\mathcal{O}((MN)^2)$
Recovery of (s_1, s_2) -Kronecker-supported sparse vectors		
TSSBL	$\mathcal{O}(N^2M + N^3)$	$\mathcal{O}(N^2)$
KroSBL [7]	$\mathcal{O}(N^3 + (MN)^2)$	$\mathcal{O}((MN)^2)$

space complexity. Since there is no SBL variant for the hierarchical sparsity model, we compare our method with HiHTP. Although TSSBL has slightly higher complexity than HiHTP, our TSHTP's complexity is lower, and we can also trade-off between time and space complexity since multiple SMV in (3) can be handled sequentially or in parallel. Also, HTP-based algorithms' complexity depends on true sparsity levels s_1 and s_2 . If these values are unknown, additional iterations from inaccurate thresholding may increase complexity due to suboptimal convergence.

III. UNIFIED RIP ANALYSIS FOR STRUCTURED SPARSITY MODELS

Our two-stage recovery suggests that the key factor for recovery is not the sparsity level $\|\mathbf{x}\|_0$, but the maximum sparsity level of different blocks, $\|\mathbf{x}_i\|_0$, and the number of nonzero blocks. We can leverage this formulation to unify the analysis for three sparsity models: standard, hierarchical, and Kronecker-supported sparsity. For this, we first introduce a generalized notion of RIP called the \mathcal{S} -RIP condition, where \mathcal{S} is the set of sparse vectors under a given sparsity model.

Definition 1. A matrix \mathbf{H} satisfies the \mathcal{S} -RIP, if there exists a constant $\delta \in (0, 1)$ such that

$$(1 - \delta)\|\mathbf{x}\|_2^2 \leq \|\mathbf{H}\mathbf{x}\|_2^2 \leq (1 + \delta)\|\mathbf{x}\|_2^2, \quad (4)$$

holds for all vectors $\mathbf{x} \in \mathcal{S}$. The smallest feasible δ , denoted as $\delta_{\mathcal{S}}(\mathbf{H})$, is the \mathcal{S} -RIC of \mathbf{H} .

Under our models, \mathcal{S} is a union of subspaces. So, our \mathcal{S} -RIP is related to bi-Lipschitz condition in [21], and hence, the theoretical results can be used to guarantee the success of compressed sensing algorithms, such as iterative hard thresholding in [10], [21]. We skip the details here, but see [22] for a discussion RIP-based conditions and structured sparsity.

By changing \mathcal{S} , we derive the different sparsity models. For example, if \mathcal{S} is the set for all s -sparse vectors, it reduces to the standard s -RIP condition. We denote the standard RIC of a given matrix \mathbf{H} as $\delta_s(\mathbf{H})$. The next result presents an upper bound for $\delta_{\mathcal{S}}(\mathbf{H})$, based on the standard RICs of \mathbf{H}_1 and \mathbf{H}_2 .

Theorem 1. For the Kronecker-structured dictionary $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$ and set $\mathcal{S} \subseteq \mathbb{R}^N$, the \mathcal{S} -RIC of \mathbf{H} satisfies

$$\delta_{\mathcal{S}}(\mathbf{H}) \leq \sup_{\mathbf{x} \in \mathcal{S}} (1 + \delta_{s_1(\mathbf{x})}(\mathbf{H}_1))(1 + \delta_{s_2(\mathbf{x})}(\mathbf{H}_2)) - 1,$$

where for any vector $\mathbf{x} \in \mathcal{S}$, the term $s_1(\mathbf{x})$ is the number of nonzero blocks in \mathbf{x} when we partition into N_1 blocks with each length N_2 , and $s_2(\mathbf{x}) = \max_i \|\mathbf{x}_i\|_0$ where \mathbf{x}_i represents the i th block of \mathbf{x} .

Proof. For any $\mathbf{x} \in \mathcal{S}$, we note that (4) bounds $\|\mathbf{H}\mathbf{x}\|_2^2 = \|\mathbf{H}_1(\mathbf{H}_2\mathbf{X})^\top\|_F^2$, where $\mathbf{x} = \text{vec}(\mathbf{X})$ and the i th column of \mathbf{X} is \mathbf{x}_i . We look into $\mathbf{H}_2\mathbf{X}$ first. Since there are only $s_1(\mathbf{x})$ among N_1 blocks of \mathbf{x} are nonzeros, the matrix $\mathbf{H}_2\mathbf{X}$ has at most $s_1(\mathbf{x})$ nonzero columns. Hence, every column of $(\mathbf{H}_2\mathbf{X})^\top$ has at most $s_1(\mathbf{x})$ non-zero entries. Using the standard RIC of \mathbf{H}_1 ,

$$(1 - \delta_{s_1(\mathbf{x})})\|\mathbf{H}_2\mathbf{X}\|_F^2 \leq \|\mathbf{H}_1(\mathbf{H}_2\mathbf{X})^\top\|_F^2 \leq (1 + \delta_{s_1(\mathbf{x})})\|\mathbf{H}_2\mathbf{X}\|_F^2. \quad (5)$$

Further, since there are at most $s_2(\mathbf{x})$ non-zero entries in each column of \mathbf{X} , we derive

$$(1 - \delta_{s_2(\mathbf{x})})\|\mathbf{X}\|_F^2 \leq \|\mathbf{H}_2\mathbf{X}\|_F^2 \leq (1 + \delta_{s_2(\mathbf{x})})\|\mathbf{X}\|_F^2. \quad (6)$$

Combining (5) and (6), we conclude

$$\begin{aligned} (1 - \delta_{s_1(\mathbf{x})})(1 - \delta_{s_2(\mathbf{x})})\|\mathbf{X}\|_F^2 &\leq \|\mathbf{H}_1(\mathbf{H}_2\mathbf{X})^\top\|_F^2 \\ &\leq (1 + \delta_{s_1(\mathbf{x})})(1 + \delta_{s_2(\mathbf{x})})\|\mathbf{X}\|_F^2. \end{aligned}$$

Since $\|\mathbf{H}\mathbf{x}\|_2^2 = \|\mathbf{H}_1(\mathbf{H}_2\mathbf{X})^\top\|_F^2$ and $\|\mathbf{x}\|_2^2 = \|\mathbf{X}\|_F^2$, we get

$$\begin{aligned} \delta_{\mathcal{S}}(\mathbf{H}) &\leq \sup_{\mathbf{x} \in \mathcal{S}} \max\{1 - (1 - \delta_{s_1(\mathbf{x})})(1 - \delta_{s_2(\mathbf{x})}), \\ &\quad (1 + \delta_{s_1(\mathbf{x})})(1 + \delta_{s_2(\mathbf{x})}) - 1\} \\ &= \sup_{\mathbf{x} \in \mathcal{S}} (1 + \delta_{s_1(\mathbf{x})})(1 + \delta_{s_2(\mathbf{x})}) - 1, \end{aligned}$$

which completes the proof. \square

We next derive the RICs for the standard s -sparsity, (s_1, s_2) -hierarchical sparsity, and (s_1, s_2) -Kronecker-supported sparsity by changing the definitions of \mathcal{S} in Theorem 1 and compare them with the existing results. We start with the standard sparsity model where \mathcal{S} is the set of s -sparse vectors.

Corollary 1. For the sparsity level s , the s -RIC of a Kronecker-structured matrix $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$ can be bounded as

$$\delta_s(\mathbf{H}) \leq \max_{1 \leq s_1 \leq s} (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s+1-s_1}(\mathbf{H}_2)) - 1.$$

Proof. We derive the result from Theorem 1 by setting \mathcal{S} as the set of s -sparse vectors. Using the notation used in Theorem 1, for any s -sparse vector \mathbf{x} , if $\max_i \|\mathbf{x}_i\|_0 = s_2(\mathbf{x})$, then there are at most $s - s_2(\mathbf{x}) + 1$ nonzero blocks in \mathbf{x} . We hence have $s_1(\mathbf{x}) \leq s - s_2(\mathbf{x}) + 1$. Thus, $\delta_s(\mathbf{H})$ can be bound by the maximum value of $(1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s_2}(\mathbf{H}_2)) - 1$ over all possible options of s_1 and s_2 satisfying the constraint, $s_1(\mathbf{x}) + s_2(\mathbf{x}) \leq s + 1$. Consequently, we arrive at the desired result. \square

Our bound in Corollary 1 is no worse than the existing bound for $\delta_s(\mathbf{H})$ in the literature [17] because

$$\begin{aligned} &\max_{1 \leq s_1 \leq s} (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s+1-s_1}(\mathbf{H}_2)) - 1 \\ &\leq \max_{1 \leq s_1 \leq s} (1 + \delta_{s_1}(\mathbf{H}_1)) \max_{1 \leq s_1 \leq s} (1 + \delta_{s+1-s_1}(\mathbf{H}_2)) - 1 \\ &\leq (1 + \delta_s(\mathbf{H}_1))(1 + \delta_s(\mathbf{H}_2)) - 1, \end{aligned} \quad (7)$$

since $\delta_s(\cdot)$ is a non-decreasing function of s and (7) is the existing bound. Further, if $\mathbf{H}_1 = \mathbf{H}_2$, we have

$$\begin{aligned} & \max_{1 \leq s_1 \leq s} (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s+1-s_1}(\mathbf{H}_1)) - 1 \\ &= \max_{1 \leq s_1 \leq \lceil s/2 \rceil} (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s+1-s_1}(\mathbf{H}_1)) - 1 \\ &\leq (1 + \delta_{\lceil s/2 \rceil}(\mathbf{H}_1))(1 + \delta_s(\mathbf{H}_1)) - 1, \end{aligned}$$

making our bound better than the existing bound.

Corollary 1 also corroborates that that only the number of nonzero blocks and the maximum number of nonzeros in each block $\max_i \|\mathbf{x}_i\|_0$ affect the s -RIC of Kronecker-structured \mathbf{H} . The intra-block measuring of \mathbf{H}_2 and the block-level measuring of \mathbf{H}_1 collectively contribute to the deviation $\|(\mathbf{H}_1 \otimes \mathbf{H}_2)\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2$, reflected by the coefficients δ_{s_1} and δ_{s_2} in Corollary 1.

Next, we look at the hierarchical sparsity model, by choosing \mathcal{S} as the set of all (s_1, s_2) -hierarchically sparse vectors.

Corollary 2. *Consider the Kronecker-structured dictionary $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$. For (s_1, s_2) -hierarchically sparse vectors, the (s_1, s_2) -RIC of \mathbf{H} , i.e., $\delta_{(s_1, s_2)}(\mathbf{H})$, can be bounded as*

$$\delta_{(s_1, s_2)}(\mathbf{H}) \leq (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s_2}(\mathbf{H}_2)) - 1,$$

We note that this result is identical to the result in [10]. Further, the RIC of Kronecker-structured \mathbf{H} for (s_1, s_2) -Kronecker-supported sparse vector \mathbf{x} is as follows.

Corollary 3. *Consider the Kronecker-structured dictionary $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$. For Kronecker-supported sparse vector \mathbf{x} where there are s_1 nonzero blocks with each s_2 -sparse, the Kronecker-supported-RIC of \mathbf{H} can be bounded as*

$$\delta_{s_1, s_2}^{\text{KS}}(\mathbf{H}) \leq (1 + \delta_{s_1}(\mathbf{H}_1))(1 + \delta_{s_2}(\mathbf{H}_2)) - 1.$$

Surprisingly, the upper bound of $\delta_{s_1, s_2}^{\text{KS}}$ matches $\delta_{(s_1, s_2)}$, despite the Kronecker-supported sparse vector \mathbf{x} being more “structured” than the (s_1, s_2) -hierarchically sparse vector. To clarify, we recall the additional structure in the Kronecker-supported sparsity model arises from the joint sparsity of the nonzero blocks of \mathbf{x} , aligning (3) with the MMV model. However, for $\mathbf{H}_2\mathbf{x}$, the (s_1, s_2) -hierarchically sparse vector does not necessarily maintain the same support across nonzero blocks, making (3) a collection of SMV problems. As noted in [22], [23], RIP analysis considers worst-case performance and does not guarantee that the standard MMV measurement model outperforms the SMV case. Consequently, our bound for δ^{KS} offers no improvement, and to the best of our knowledge, establishing a stronger RIP-based condition for the MMV model remains an open problem in the literature.

IV. NUMERICAL EVALUATIONS

In this section, we present numerical results for the recovery of regular s -sparse vector, (s_1, s_2) -hierarchically sparse vector, and (s_1, s_2) -Kronecker-supported sparse vector. We use SBL and MMV-SBL [19] for the compressed sensing algorithm in Algorithm 1. Our benchmarks for the standard sparsity model are SBL [18] (state-of-the-art) and OMP. We use HiHTP

[10] (state-of-the-art), SBL, and OMP as benchmarks for the hierarchically sparse vector recovery. For (s_1, s_2) -Kronecker-structured support sparse vector, we benchmark with AM-KroSBL and SVD-KroSBL (state-of-the-art) [7].

Our setting is as follows. For the Kronecker-structured dictionary $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$, we set $M_1 = M_2 = 30$ and $N_1 = N_2 = 40$. The entries of \mathbf{H}_1 and \mathbf{H}_2 and nonzero entries of \mathbf{x} are drawn independently from the standard normal distribution. For s -sparse vectors, we set $s = 15$, and the support is randomly drawn from a uniform distribution. For (s_1, s_2) -hierarchically and (s_1, s_2) -Kronecker-supported sparse vectors, we opt for $s_1 = s_2 = 5$. Here, supports are generated by first selecting s_1 blocks uniformly at random, then assigning support within each block uniformly. In the hierarchical sparsity model, support varies across blocks, whereas in the Kronecker-supported sparsity model, it remains identical. We adopt the additive white Gaussian noise with zero mean whose variance is determined by $\text{SNR (dB)} = 10 \log_{10} \mathbb{E}\{\|\mathbf{H}\mathbf{x}\|_2^2 / \|\mathbf{n}\|_2^2\}$ of $\{0, 5, 10, 15, 20, 25\}$. Two metrics are considered for performance evaluation: normalized mean squared error (NMSE) and run time. Here, we define $\text{NMSE} = \mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 / \|\mathbf{x}\|_2^2\}$, where \mathbf{x} is the ground truth and $\hat{\mathbf{x}}$ is the estimated vector. We limit the number of iterations for all SBL-based methods (TSSBL, SBL, MMV-SBL, AM-KroSBL, and SVD-KroSBL) to three hundred. We also prune small entries in hyperparameters for faster convergence for SBL-based algorithms. Simulation results, summarized in Fig. 1 and Table II, are averaged over 500 independent trials.

Fig. 1 compares the performance of different algorithms under each sparsity model. In Fig. 1(a), we compare our TSSBL with SBL and OMP. Our TSSBL outperforms OMP in all cases and outperforms SBL in low SNR. The worse performance of TSSBL in high SNR cases is because the algorithm can ignore some blocks of \mathbf{x} having fewer nonzero entries in the first step when the noise is present. The worse performance of SBL in low SNR case is because the noise estimation step in SBL is not robust in strong noise case [24] and can lead to degraded performance. We then compare our TSSBL with SBL, OMP, and HiHTP in Fig. 1(b). We note that HiHTP requires the true sparsity levels s_1 and s_2 as input, which are not always known in real applications. Thus, for a fair comparison, we do not input the true sparsity level but only a rough approximation. Due to the thresholding operator, HiHTP can effectively enforce the hierarchical sparsity, making the algorithm more robust against strong noise. However, when SNR increases, the drawback of requiring accurate sparsity level prevails, hence our TSSBL dominates. The comparison between our TSSBL and AM-/SVD-KroSBL is shown in Fig. 1(c), where it shows that our TSSBL is able to perform similarly or better for all SNR values.

Furthermore, from Table II, we see that the run time our TSSBL is at least one order less than the other candidates. We attribute this to exploiting the Kronecker structure of \mathbf{H} . Overall, our TSSBL offers comparable performance with a considerably shorter run time than the state-of-the-art methods.

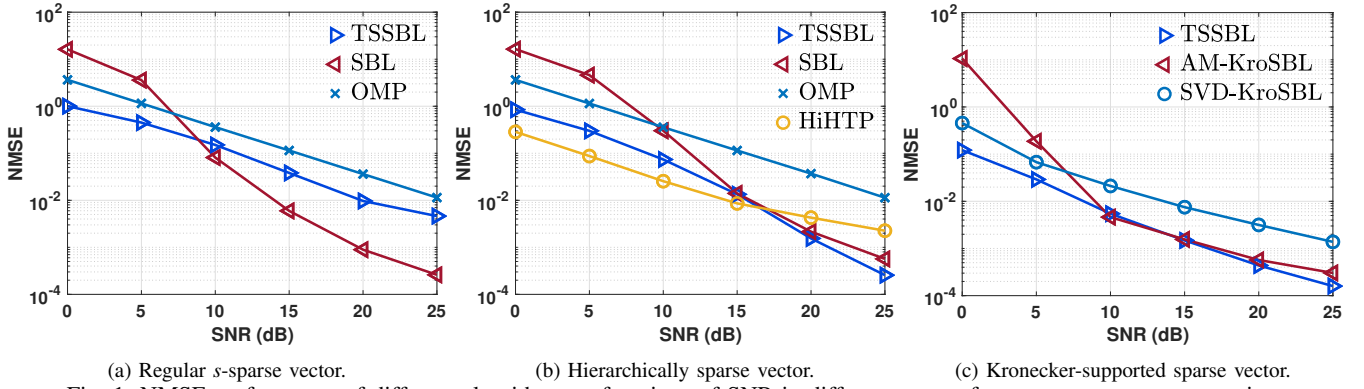


Fig. 1: NMSE performance of different algorithms as functions of SNR in different types of sparse vector recovery setting.

TABLE II: Runtime in seconds. **Bold:** the best result.

SNR	0 dB	5 dB	10 dB	15 dB	20 dB	25 dB
Recovery of s -sparse vectors						
TSSBL	0.122	0.082	0.040	0.022	0.015	0.010
OMP	0.606	0.604	0.598	0.602	0.600	0.600
SBL	3.676	4.174	2.526	1.233	0.986	0.923
Recovery of (s_1, s_2) -hierarchically sparse vectors						
TSSBL	0.092	0.046	0.015	0.009	0.006	0.004
OMP	0.606	0.606	0.599	0.599	0.598	0.598
SBL	3.559	3.814	3.299	1.387	0.977	0.873
HiHTP	0.821	0.830	0.840	0.812	0.826	0.808
Recovery of (s_1, s_2) -Kronecker-supported sparse vectors						
TSSBL	0.028	0.015	0.004	0.002	0.001	0.001
AM-KroSBL	7.572	8.145	3.990	2.142	1.263	0.815
SVD-KroSBL	4.299	1.386	0.530	0.356	0.308	0.292

V. CONCLUSION

This work focused on the Kronecker compressed sensing problem with multiple sparsity structures. We first explored the hierarchical view of the Kronecker-structured dictionary. Each factor matrix in the Kronecker-structured dictionary measures the sparse signal at different levels. Based on the hierarchical view, we developed a two-stage sparse recovery algorithm, which offers comparable performance compared with other state-of-the-art algorithms with lower computational complexity. We then unified the RIP analysis of Kronecker-structured matrix with different structured sparsity models. Designing new algorithms through hierarchical view and establishing recovery guarantees are exciting avenues for future research.

REFERENCES

- [1] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. Image Process.*, vol. 21, pp. 494–504, Aug. 2011.
- [2] C. F. Caiafa and A. Cichocki, "Block sparse representations of tensors using Kronecker bases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal, Process.*, pp. 2709–2712, Mar. 2012.
- [3] C. F. Caiafa and A. Cichocki, "Computing sparse representations of multidimensional signals using Kronecker bases," *Neural Comput.*, vol. 25, pp. 186–220, Jan. 2013.
- [4] C. F. Caiafa and A. Cichocki, "Multidimensional compressed sensing and their applications," *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, vol. 3, pp. 355–380, Oct. 2013.
- [5] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, pp. 1524–1538, July 2017.
- [6] R. Wang, H. Ren, C. Pan, G. Zhou, and J. Wang, "Tensor decomposition-based time varying channel estimation for mmWave MIMO-OFDM systems," *arXiv preprint arXiv:2403.02942*, 2024.
- [7] Y. He and G. Joseph, "Bayesian algorithms for Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation," *IEEE Trans. Signal Process.*, June 2024.
- [8] S. Friedland, Q. Li, and D. Schonfeld, "Compressive sensing of sparse tensors," *IEEE Trans. Image Process.*, vol. 23, pp. 4438–4447, Oct. 2014.
- [9] Y. He and G. Joseph, "Structure-aware sparse Bayesian learning-based channel estimation for intelligent reflecting surface-aided MIMO," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal, Process.*, pp. 1–5, Jun. 2023.
- [10] I. Roth, M. Kliesch, A. Flinth, G. Wunder, and J. Eisert, "Reliable recovery of hierarchically sparse signals for Gaussian and Kronecker product measurements," *IEEE Trans. Signal Process.*, vol. 68, pp. 4002–4016, June 2020.
- [11] X. Xu, S. Zhang, F. Gao, and J. Wang, "Sparse Bayesian learning based channel extrapolation for RIS assisted MIMO-OFDM," *IEEE Trans. Commun.*, vol. 70, pp. 5498–5513, Aug. 2022.
- [12] Y. He and G. Joseph, "Kronecker-structured sparse vector recovery with application to IRS-MIMO channel estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal, Process.*, pp. 1–5, Apr. 2025.
- [13] M. F. Duarte and R. G. Baraniuk, "Kronecker product matrices for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal, Process.*, pp. 3650–3653, Mar. 2010.
- [14] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.
- [15] W.-C. Chang and Y. T. Su, "Sparse Bayesian learning based tensor dictionary learning and signal recovery with application to MIMO channel estimation," *IEEE J. Sel. Top. Signal Process.*, vol. 15, pp. 847–859, Apr. 2021.
- [16] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, pp. 21–30, Mar. 2008.
- [17] M. F. Duarte and R. G. Baraniuk, "Kronecker product matrices for compressive sensing," *Rice University, Houston, Tech. Rep.*, Mar. 2011.
- [18] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, pp. 2153–2164, Aug. 2004.
- [19] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, pp. 3704–3716, June 2007.
- [20] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy algorithms for joint sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, pp. 1694–1704, Jan. 2014.
- [21] T. Blumensath, "Sampling and reconstructing signals from a union of linear subspaces," *IEEE Trans. Inf. Theory*, vol. 57, pp. 4660–4671, July 2011.
- [22] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, pp. 5302–5316, Oct. 2009.
- [23] B. Li and A. P. Petropulu, "Structured sampling of structured signals," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, pp. 1009–1012, IEEE, Feb. 2013.
- [24] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Top. Signal Process.*, vol. 5, pp. 912–926, Sept. 2011.